# NSARM: Next-Scale Autoregressive Modeling for Robust Real-World Image Super-Resolution

**Xiangtao Kong**[1,2], **Rongyuan Wu**[1,2], **Shuaizheng Liu**[1,2], **Lingchen Sun**[1,2], **Lei Zhang**[1,2] *
[1]The Hong Kong Polytechnic University [2]OPPO Research Institute
{xiangtao.kong, rongyuan.wu, Shuaizheng.liu, lingchen.sun}@connect.polyu.hk
Project page: https://github.com/Xiangtaokong/NSARM

## Abstract

Most recent real-world image super-resolution (Real-ISR) methods employ pre-trained text-to-image (T2I) diffusion models to synthesize the high-quality image either from random Gaussian noise or directly from the input low-quality image. These approaches train ControlNet or LoRA modules while keeping the pre-trained model fixed, which often introduces over-enhanced artifacts and hallucinations, suffering from limited robustness to inputs with varying degradations. Recent visual autoregressive (AR) models, such as pre-trained *Infinity*, can provide strong T2I generation capabilities while offering superior efficiency by using the bitwise next-scale prediction strategy. Building upon next-scale prediction, we introduce a robust Real-ISR framework with a generation pathway control mechanism, namely Next-Scale Autoregressive Modeling (NSARM). Specifically, we train NSARM in two stages: a transformation network is first trained to map the input low-quality image to preliminary scales, followed by an end-to-end full-model AR fine-tuning. Such a comprehensive fine-tuning enhances the robustness of NSARM in Real-ISR tasks without compromising its generative capability. Extensive quantitative and qualitative evaluations demonstrate that as a pure AR model, NSARM achieves superior visual results over existing Real-ISR methods, with fast inference speed and comparable fidelity. Most importantly, it demonstrates much higher robustness to varying input quality, showing stronger generalization performance.

## 1 Introduction

Real-world image super-resolution (Real-ISR) focuses on reconstructing high-resolution (HR) images from their low-resolution (LR) counterparts degraded by unknown real-world distortions. Unlike traditional ISR methods [6, 7, 5, 4], which focus on specific degradations and optimize MSE-based losses, Real-ISR emphasizes perceptual quality. Generative approaches such as generative adversarial networks (GANs) [10] have been widely adopted for this task. Although GAN-based methods [14, 30, 42] can improve the visual perception of ISR output, they often suffer from training instability and generate many visual artifacts.

The success of multimodal foundation models [21, 22] and text-to-image (T2I) diffusion models (*e.g.*, Stable Diffusion (SD) [19, 8]) pretrained on vast image data has enabled significant progress in Real-ISR. Leveraging the strong generative priors of these models, many SD-based approaches have significantly surpassed GAN-based methods in visual quality. With LR images as condition, some works [34, 37, 36] synthesize HR images by iteratively denoising random Gaussian noise at higher resolution, yet they require numerous diffusion steps and suffer from slow inference. In contrast, some methods [33, 24] directly generate HR images from LR inputs with fewer diffusion

*Corresponding author (e-mail: cslzhang@comp.polyu.edu.hk)
This research is supported by the PolyU-OPPO Joint Innovative Research Center.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

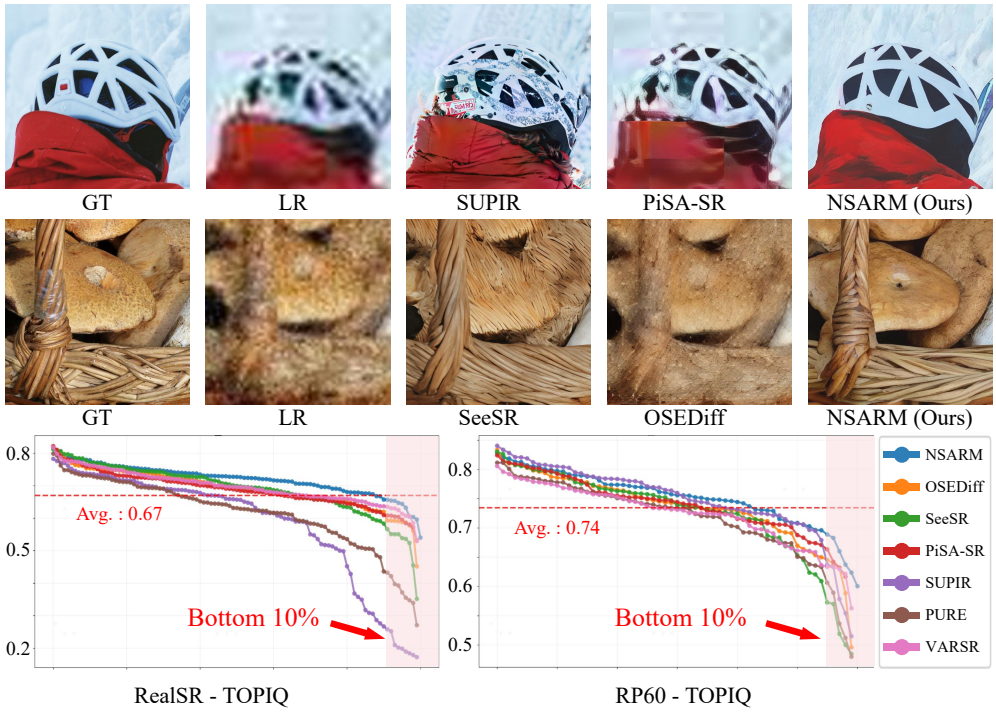



Figure 1: **Top two rows**: failure cases of existing Real-ISR methods, while our NSARM still works. **Bottom row**: sorted distributions of TOPIQ scores of competing methods on RealSR and RP60 datasets. We see that the quality curves of existing methods fall sharply in the late portion, indicating failure cases. For some methods, more than 10% of the cases can fail. Our method demonstrates significantly better robustness than existing methods.

steps, greatly improving efficiency but at the cost of reduced generative capability. Meanwhile, most of the existing methods rely on ControlNet or LoRA modules to adapt a pre-trained T2I model to the Real-ISR task, often introducing over-enhanced artifacts and hallucinations, and reducing the robustness to the varying degradations of the LR input (see Fig. 1 for examples). Failure cases can occur with a probability of 10% to 20% for some methods (see the bottom row of Fig. 1 and Fig. 5 for details). Furthermore, if we attempt to fine-tune the pre-trained SD model using task-specific losses to improve the robustness, the generative capability of the model will be degraded significantly.

Recent advances in visual autoregressive (AR) models for T2I generation offer potential solutions to the aforementioned challenges. Instead of executing the diffusion process in high-resolution space, AR-based methods generate images by predicting visual tokens one-by-one [9, 16]; however, these approaches are not well suited for Real-ISR tasks. For example, PURE [31] applies this token-by-token generation to Real-ISR but leads to a very slow inference speed (see Tab. 2). Different from SD or other AR methods, VAR [25] introduces a next-scale prediction approach, which learns to predict residual information between different image scales, progressively refining the image from low to high resolution. This coarse-to-fine framework naturally matches the requirement of Real-ISR tasks, which model image relationships between different resolution levels. VARSR [20] adapts VAR for Real-ISR with this paradigm. However, VARSR relies on a discrete pre-trained codebook, which fundamentally constrains the quality of reconstructed HR images. Although VARSR employs a diffusion-based post-processing module to reduce artifacts from discrete representations, this introduces additional complexity, and it is hard to obtain Real-ISR results comparable to SD-based approaches in visual quality.

To overcome the limitations of existing SD-based and AR-based Real-ISR methods, inspired by Infinity [11], we propose Next-Scale Autoregressive Modeling (NSARM) - a novel approach that performs next-scale prediction in a bitwise quantized space to progressively refine images from LR to HR. Via bitwise quantization, Infinity introduces the Infinite-Vocabulary Tokenizer, which provides powerful generative priors with a theoretically continuous vocabulary and high processing speed. As shown in Fig. 2, to utilize the benefits of pre-trained Infinity model, we introduce a key innovation in

NSARM: Real-ISR generation pathway control – a transformation network that directly processes the LR image to generate the preliminary few scales. By replacing the original scales, which could be further refined by Infinity, we could control the generation pathway (See Sec. 3 and Fig. 3). We then present a two-stage training strategy to train the NSARM model: we first train the transformation network and then fine-tune the entire AR pathway followed by optimizing a fidelity refiner. NSARM generates HR output from the LR image instead of random noise, and it is fully fine-tuned by the original pre-training objective and loss function; therefore, it is robust to the input LR image without sacrificing the generation capability. As can be seen from the bottom rows of Fig. 1 and Fig. 5, NSARM shows much stronger robustness than the existing methods in Real-ISR.

Our contributions are summarized as follows. First, we propose NSARM, an efficient AR model enabled by preliminary scale transformation and bitwise next-scale prediction. Second, our proposed Real-ISR generation pathway control and two-stage training support both generation directly from LR input and full-parameter fine-tuning, enabling exceptional model robustness. Finally, NSARM demonstrates robust and superior visual results compared to existing SD-based and AR-based approaches, while maintaining fast inference speed and comparable fidelity, demonstrating outstanding potential for the Real-ISR task.

## 2 Related Work

**Real-World Image Super-Resolution.** Traditional ISR methods [6, 7, 5, 4] typically focus on limited degradation types while optimizing with MSE-based losses, often yielding over-smoothed results with poor generalization to real-world scenarios. Driven by the growing demands for perceptual excellence in real-world deployment, recent works aim to synthesize realistic degradations to train Real-ISR models. Some real-world ISR datasets have been developed [2, 32], and RealESRGAN [29] and BSRGAN [40] introduce degradation pipelines that simulate real-world conditions (*e.g.*, noise, blur, and compression artifacts), enabling single models to handle diverse degradations while improving generalization. In this work, we adopt the degradation settings of RealESRGAN to implement the model training.

**Diffusion-based Real-ISR Models.** While GAN-based Real-SR approaches [14, 30, 42] can achieve improved visual quality, they suffer from training instability and limited natural image prior due to insufficient pre-training. Stable Diffusion (SD) models [19, 8] could provide much stronger priors due to their large-scale T2I pre-training, catalyzing diverse studies. Some works directly harness these priors to boost reconstruction quality [28, 15, 36], and others explore text prompting for better semantic guidance [34, 37]. These methods synthesize the high-quality image from random Gaussian noise, which yields realistic results but is slow. Methods have also been developed to accelerate the Real-ISR process by reducing diffusion steps [33, 39, 24], but visual quality is degraded due to the reduction in generation diversity. Most SD-based approaches employ generative priors by fixing the pre-trained diffusion model. However, these methods lack enough robustness and may introduce severe artifacts when handling images of varying contents and degradations.

**AR-based Real-ISR Models.** Early approaches like VQGAN [9] and Parti [38] employ quantization to convert images into discrete tokens for next-token prediction. PURE [31] introduces this paradigm to Real-ISR, but it suffers from slow inference speed. VAR [25] proposes next-scale prediction and significantly improves the sampling speed. Based on the VAR model, VARSR [20] obtains fast inference speed for the Real-ISR task. However, these methods require discrete codebooks, which fundamentally limit their reconstruction capability. The Infinity [11] model uses bitwise modeling to replace index-wise tokens with bitwise tokens, which could theoretically scale the tokenizer vocabulary to infinity. It also offers efficient token prediction, making it possible to train on a large scale. Consequently, Infinity provides more powerful generative priors with nearly continuous representations, making it well-suited for Real-ISR tasks.

## 3 Method

### 3.1 Bitwise Visual Autoregressive Model

**Next-Scale Prediction.** Traditional visual AR models first encode and decompose a GT image $\boldsymbol{I}$ to image tokens $(x_1, x_2, ..., x_T)$, then learn to predict the next token one by one from previous tokens

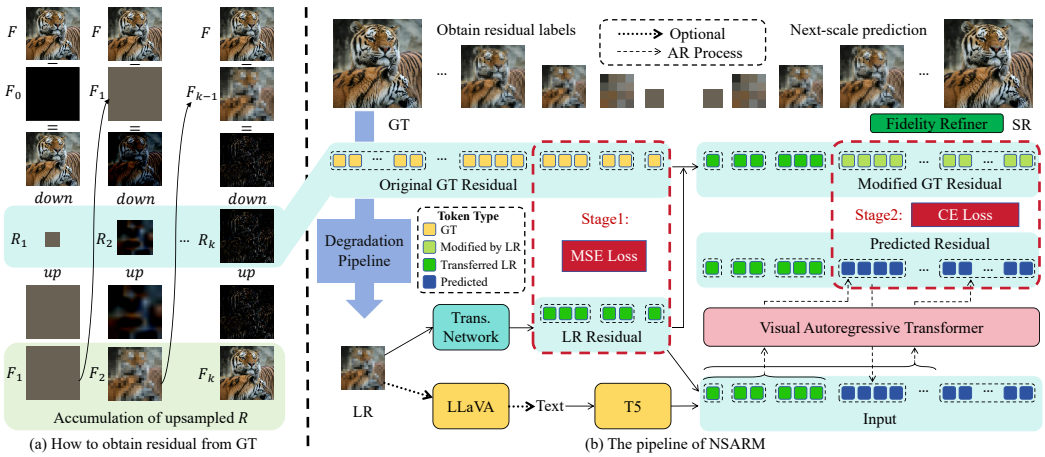



Figure 2: (a) The image decomposition process of VAR-like methods. (b) The overview framework of our proposed NSARM.

with condition $c$:

$$p\left(x_1, x_2, ..., x_T\right) = \prod_{t=1}^{T} p\left(x_t \mid x_1, x_2, ..., x_{t-1}, c\right). \tag{1}$$

As shown in Fig. 2, VAR-like models first encode the image $\boldsymbol{I}$ into latent features $\boldsymbol{F} \in \mathbb{R}^{h \times w \times d}$ and then decompose the features $\boldsymbol{F}$ into $K$ multi-scale residual maps $(\boldsymbol{R}_1, \boldsymbol{R}_2, ..., \boldsymbol{R}_K)$ by:

$$\begin{aligned} \boldsymbol{R}_k &= \text{down}(\boldsymbol{F} - \boldsymbol{F}_{(k-1)}, (h_k, w_k)), \\ \boldsymbol{F}_k &= \sum_{i=1}^{k} \text{up}(\boldsymbol{R}_i, (h, w)), \end{aligned} \tag{2}$$

where $\boldsymbol{F}_k$ is accumulated from $\boldsymbol{R}_1$ to $\boldsymbol{R}_k$ ($\boldsymbol{F}_0$ is zero), $\text{up}(\cdot)$ and $\text{down}(\cdot)$ mean upsampling and downsampling. The resolution of $\boldsymbol{R}_k$ is $h_k \times w_k$ and it grows gradually from $k = 1 \rightarrow K$. Since each $\boldsymbol{R}_k$ is obtained by the GT feature $\boldsymbol{F}$ subtracting the accumulated $\boldsymbol{F}_{k-1}$, there is no information loss in the decomposition. These residual maps are used as the input and target during training (see our pseudo codes of Alg.A.1 in the **Appendix A**). Then, VAR models learn to predict residuals $\boldsymbol{R}_k$ of the next scale conditioned on previous ones $(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_{k-1})$ and the condition $c$:

$$p(\boldsymbol{R}_1, ..., \boldsymbol{R}_K) = \prod_{k=1}^{K} p(\boldsymbol{R}_k \mid \boldsymbol{R}_1, ..., \boldsymbol{R}_{k-1}, c). \tag{3}$$

**Infinite-Vocabulary Tokenizer and Classifier.** Infinity [11] adopts binary spherical quantization (BSQ) [43] to increase the vocabulary size to an extremely large scale, *e.g.*, $2^{64}$. Specifically, BSQ quantizes the continuous $\boldsymbol{R}_k \in (h_k, w_k, d)$ of Eq. 2 to binary. In this way, all possible binary combinations form a vast vocabulary of size $2^d$, which is nearly continuous. For instance, the commonly used representation of Float16 corresponds to $2^{16}$ combinations, which is even smaller than this vocabulary. Consequently, the infinite-vocabulary tokenizer can achieve reconstruction results comparable to the continuous encoding of SD. Consequently, Infinity learns to predict binary residual tokens (bitwise 0/1 indices) for the next scale through cross-entropy optimization. For a $2^d$-sized vocabulary, the infinite-vocabulary classifier (IVC) replaces a single $2^d$-class classifier with $d$ parallel binary classifiers, reducing the parameter count from $O(2^d)$ to $O(d)$.

### 3.2 The Pipeline of NSARM

The pipeline of NSARM comprises two main components: (1) the Real-ISR generation pathway control mechanism, and (2) a two-stage optimization protocol.

**Real-ISR Generation Pathway Control.** During inference, Infinity generates high-resolution images through progressive scale prediction. For a target image resolution of 1024×1024, there will be 13 discrete scales with resolution $\{16, 32, 64, 96, 128, 192, 256, 320, 384, 512, 640, 768, 1024\}$. Through systematic experimentation, we make two important observations about the process.

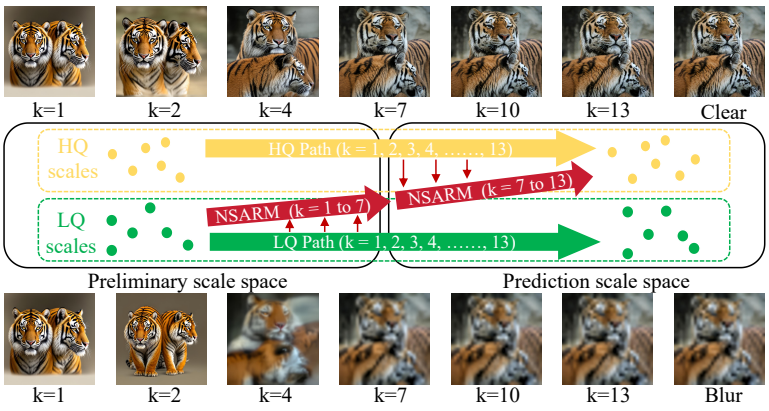



Figure 3: The top and bottom images are the generation results of Infinity with the $k$ scales replaced by clear or blurred images, respectively, corresponding to the HQ and LQ pathways. NSARM is to establish a pathway towards the desired HQ by preliminary scales from the LR input.

First, as depicted in the top row of Fig. 3, replacing the preliminary $k$ predicted scales with the corresponding decomposed residuals from a clear image (Eq. 2, with prompt: "Two tigers, one from the front and one from the side." ), it progressively steers the output towards the clear image with the increase of $k$. In contrast, smaller $k$ values introduce greater generation randomness. Pure T2I generation tasks correspond to $k = 1$, while full replacement ($k = 13$) yields the original image. The LR input provides information for the preliminary scales so that the next-scale prediction has the potential to handle the Real-ISR task.

Second, the bottom row of Fig. 3 demonstrates that even replacing only the earlier scales (*e.g.*, $k = 7$) with residuals derived from a blurry input, the output remains blurry, despite the model's powerful generative prior. Similarly, if preliminary scales contain degradation artifacts, they would lead to amplified artifacts in the final image. This indicates that the preliminary scales establish a critical "generation pathway" that guides subsequent synthesis.

As shown in Fig. 3, the key insight of our NSARM is to establish a faithful pathway towards the desired high-quality output by preliminary scales from the LR input. Our method can make better use of Infinity's powerful generative prior to achieve high-quality restoration. To establish the pathway control, we propose a dedicated two-stage optimization protocol.

**Stage 1: Pathway-Alignment by Transformation Network.** This stage trains a lightweight transformation network. To demonstrate the effectiveness of the pipeline instead of relying on a meticulously designed transformation network, we intentionally design the network to be simple, employing a 3-layer CNN for LR feature extraction and a scale-specific transformer (4–16 layers/heads, smaller for coarser scales) for linear projection to the AR intermediate size (Fig. 4). It maps the input LR image into the preliminary scale residuals required by Infinity's image generation pathway. For 4× super-resolution targeting 1024×1024 resolution (LR input: 256×256), the transformation network outputs the first seven scales of the progressive sequence ($\{16, 32, 64, 96, 128, 192, 256\}$). This design ensures: (1) full retention of LR information without spatial compression and (2) delegating preliminary degradation removal at the LR level to the transformation network, enabling the autoregressive process to focus on refined detail synthesis over subsequent scales. We also show the ablation study of $k$ in the **Appendix B**.

As illustrated in Fig. 2, we first train the transformation network independently using MSE loss in the feature domain to align the HR generation pathway from the decomposed GT:

$$\mathcal{L}_{s_1} = \frac{1}{K_t} \sum_{k=1}^{K_t} \|\boldsymbol{R}_k - \mathcal{T}(I_{LR})_k\|_2^2, \tag{4}$$

where $K_t = 7$, $\mathcal{T}(\cdot)$ denotes the transformation network, $I_{LR}$ is the LR input, and $\boldsymbol{R}_k$ are the GT residuals for scale $k$. The network output cannot be perfectly aligned with the subsequent autoregressive process. While this stage provides basic preliminary scales, the ablation experiments in Sec. B.1 reveal that using only stage 1 training will introduce artifacts.

**Stage 2: Next-Scale Autoregressive Fine-tuning.** As shown in Fig. 2, we then fine-tune the whole NSARM model, including the visual autoregressive transformer, given the transformed preliminary LR scales $(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_{K_t})$. The next-scale prediction can be formulated as follows:

$$p(\boldsymbol{R}_{K_t+1}, ..., \boldsymbol{R}_K) = \prod_{k=K_t+1}^{K} p(\boldsymbol{R}_k \mid \boldsymbol{R}_1, ..., \boldsymbol{R}_{k-1}, c), \tag{5}$$

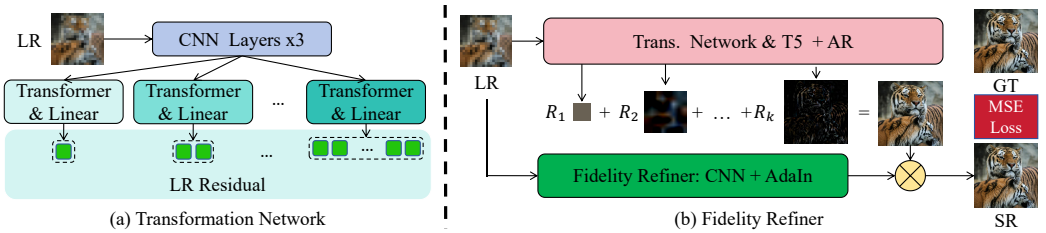



Figure 4: The details of (a) Transformation Network and (b) Fidelity Refiner.

where $K_t = 7$, and $c$ is the description text of the LR image. Following Infinity's pre-training, we use bitwise cross-entropy loss to supervise binary token predictions, addressing the large token space in high-resolution scale prediction:

$$\mathcal{L}_{AR} = -\frac{1}{N}\sum_{i=1}^{N}\left[\mathbf{r}_i^{\text{MGT}}\log p(\mathbf{r}_i^{\text{pred}}) + (1-\mathbf{r}_i^{\text{MGT}})\log(1-p(\mathbf{r}_i^{\text{pred}}))\right], \tag{6}$$

where $N = h_k \times w_k$ denotes the total tokens on scale $k$, $\mathbf{r}_i^{\text{MGT}} \in \{0,1\}$ is the GT binary token which has been modified by transformed preliminary LR scales $(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_{K_t})$ through Eq. 2, and $p(\mathbf{r}_i^{\text{pred}})$ is the predicted probability of token $i$. It should be noted that the fine-tuning protocol at this stage preserves Infinity's original pre-training objective and constraints. The only modification is that the preliminary scales are now supplied by the transformation network rather than being generated from the condition text. This strategic design allows NSARM to maximally retain Infinity's inherent generative priors while adapting the generation pathway to Real-ISR requirements.

Because errors may accumulate during scale-by-scale autoregressive prediction, we introduce a lightweight fidelity refiner after the AR pathway. As shown in Fig. 4, it consists of CNN layers with AdaIN [12] and uses the LR image to refine the accumulated AR features toward the GT features under an MSE objective. In Stage 2, we optimize the AR pathway and the refiner sequentially rather than jointly. We first optimize the AR pathway with $\mathcal{L}_{AR}$ to preserve its generative prior and perceptual capability. Once this optimization is completed, we freeze the AR pathway and optimize only the refiner with $\mathcal{L}_{MSE}$. The Stage 2 optimization objective is therefore defined as

$$\mathcal{L}_{s_2} = \begin{cases} \mathcal{L}_{AR}, & \text{when optimizing the AR pathway,} \\ \mathcal{L}_{MSE}, & \text{when optimizing the refiner with the AR pathway frozen.} \end{cases} \tag{7}$$

This strategy offers a practical trade-off between perceptual quality and reconstruction fidelity: the intermediate AR output preserves richer perceptual details, while the refined output favors pixel-level fidelity. Both are obtained in a single inference run, as the former is directly available before the lightweight refinement. Detailed training schedules are provided in Sec. 4.1.

Overall, Stage 1 initializes the LR-to-residual mapping, while Stage 2 adapts the AR generation pathway and subsequently improves fidelity. As shown in Sec. B.1, omitting the Stage 1 initialization leads to very slow convergence.

# 4 Experiments

## 4.1 Experimental Settings

**Training Details.** We train NSARM for $\times 4$ Real-ISR with $1024 \times 1024$ target resolution using AdamW [18]. In the first stage, we separately train the transformation network for 100K iterations, using a learning rate of $2 \times 10^{-5}$ and batch size 196 to ensure stable convergence. In the second stage, we first optimize the AR pathway for 40K iterations using a learning rate of $4 \times 10^{-5}$ and a batch size of 32. After this optimization, we freeze the AR pathway and train only the fidelity refiner for an additional 10K iterations. The training is implemented using 16 NVIDIA A800 GPUs. Our training data comprises 1M $1024 \times 1024$ high-quality images with text annotation from Unsplash [26]. LR-GT pairs are synthesized using RealESRGAN's degradation pipeline.

**Compared Methods.** We compare NSARM with state-of-the-art SD-based and AR-based Real-ISR methods, including SeeSR [34], OSEDiff [33], PiSA-SR [24], SUPIR [37], PURE [31] and

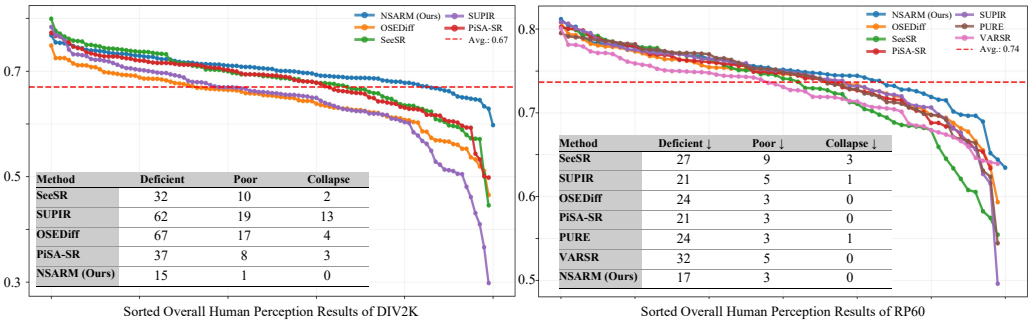
Table 1: Performance comparison of different Real-ISR methods on commonly used datasets. Red and blue indicate the best and second-best performance, respectively. In addition to the final NSARM results, we report NSARM *(w/o refiner)*, which denotes the intermediate output obtained during the same inference run. These entries are shown in gray and excluded from the red/blue ranking. Cells in these rows that would otherwise rank first are highlighted in pink.

| Datasets | Methods | PSNR↑ | SSIM↑ | LPIPS↓ | CLIPIQA | | MUSIQ | | MANIQA | | TOPIQ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Avg.↑ | Var.↓ | Avg.↑ | Var.↓ | Avg.↑ | Var.↓ | Avg.↑ | Var.↓ |
| Simulated Dataset (Evaluated on generating 1024 × 1024 images) | | | | | | | | | | | | |
| DIV2K | SeeSR [34] | 22.34 | 0.5757 | 0.3452 | 0.7074 | 0.0119 | 71.72 | 24.7735 | 0.6157 | 0.0041 | 0.6878 | 0.0068 |
| | SUPIR [37] | 21.14 | 0.5248 | 0.3784 | 0.6096 | 0.0190 | 70.19 | 61.8711 | 0.6189 | 0.0084 | 0.6266 | 0.0115 |
| | OSEDiff [33] | 22.10 | 0.5681 | 0.3389 | 0.6637 | 0.0126 | 70.43 | 25.1750 | 0.5893 | 0.0033 | 0.6177 | 0.0047 |
| | PiSA-SR [24] | 22.23 | 0.5660 | 0.3380 | 0.7018 | 0.0111 | 71.70 | 32.4888 | 0.6156 | 0.0036 | 0.6747 | 0.0045 |
| | **NSARM (ours)** | 20.93 | 0.5145 | 0.3908 | 0.7485 | 0.0053 | 74.43 | 11.4218 | 0.6157 | 0.0028 | 0.6956 | 0.0014 |
| | NSARM *(w/o refiner)* | 20.01 | 0.4900 | 0.3836 | 0.7762 | 0.0042 | 75.02 | 8.4763 | 0.6337 | 0.0027 | 0.7182 | 0.0013 |
| Real-world Datasets (Evaluated on generating 512 × 512 images) | | | | | | | | | | | | |
| DRealSR | SeeSR [34] | 28.76 | 0.7762 | 0.3316 | 0.6401 | 0.0217 | 61.11 | 174.4790 | 0.5603 | 0.0123 | 0.6217 | 0.0207 |
| | SUPIR [37] | 25.00 | 0.6373 | 0.4334 | 0.6921 | 0.0217 | 59.75 | 192.8875 | 0.5506 | 0.0120 | 0.5871 | 0.0296 |
| | OSEDiff [33] | 28.60 | 0.7872 | 0.3296 | 0.7082 | 0.0161 | 60.91 | 142.5506 | 0.5350 | 0.0089 | 0.6208 | 0.0113 |
| | PiSA-SR [24] | 28.88 | 0.7974 | 0.3145 | 0.6453 | 0.0217 | 60.33 | 131.9325 | 0.5243 | 0.0091 | 0.5987 | 0.0130 |
| | PURE [31] | 25.10 | 0.6184 | 0.4515 | 0.6694 | 0.0165 | 62.38 | 113.5275 | 0.5839 | 0.0068 | 0.5888 | 0.0108 |
| | VARSR [20] | 28.22 | 0.7660 | 0.3531 | 0.7216 | 0.0097 | 68.18 | 81.4851 | 0.6167 | 0.0049 | 0.6793 | 0.0057 |
| | **NSARM (ours)** | 28.05 | 0.7381 | 0.3919 | 0.7231 | 0.0077 | 66.68 | 92.7997 | 0.5759 | 0.0057 | 0.6933 | 0.0062 |
| | NSARM *(w/o refiner)* | 26.41 | 0.6918 | 0.4142 | 0.7337 | 0.0088 | 68.65 | 80.8570 | 0.6198 | 0.0062 | 0.7201 | 0.0052 |
| RealSR | SeeSR [34] | 25.52 | 0.7369 | 0.3065 | 0.6663 | 0.0141 | 68.57 | 44.2620 | 0.6222 | 0.0052 | 0.6908 | 0.0063 |
| | SUPIR [37] | 23.69 | 0.6632 | 0.3568 | 0.6594 | 0.0243 | 61.72 | 166.3464 | 0.5763 | 0.0094 | 0.5857 | 0.0291 |
| | OSEDiff [33] | 25.52 | 0.7334 | 0.3275 | 0.7393 | 0.0081 | 68.65 | 34.3655 | 0.6265 | 0.0037 | 0.6924 | 0.0035 |
| | PiSA-SR [24] | 25.91 | 0.7526 | 0.2902 | 0.6803 | 0.0104 | 69.14 | 29.6003 | 0.6259 | 0.0027 | 0.6885 | 0.0029 |
| | PURE [31] | 22.61 | 0.6029 | 0.3848 | 0.6909 | 0.0100 | 66.89 | 50.1819 | 0.6265 | 0.0033 | 0.6169 | 0.0116 |
| | VARSR [20] | 25.55 | 0.7263 | 0.3218 | 0.6996 | 0.0068 | 71.30 | 24.5466 | 0.6544 | 0.0028 | 0.6978 | 0.0025 |
| | **NSARM (ours)** | 24.83 | 0.6758 | 0.3655 | 0.7309 | 0.0054 | 70.45 | 27.6000 | 0.6067 | 0.0038 | 0.7204 | 0.0023 |
| | NSARM *(w/o refiner)* | 23.52 | 0.6480 | 0.3719 | 0.7382 | 0.0054 | 71.83 | 23.8346 | 0.6579 | 0.0031 | 0.7401 | 0.0017 |
| RP60 | SeeSR [34] | - | - | - | 0.7881 | 0.0184 | 71.86 | 21.8667 | 0.6654 | 0.0023 | 0.7253 | 0.0061 |
| | SUPIR [37] | - | - | - | 0.8275 | 0.0096 | 71.80 | 20.8673 | 0.6551 | 0.0045 | 0.7490 | 0.0041 |
| | OSEDiff [33] | - | - | - | 0.8424 | 0.0070 | 71.88 | 17.1801 | 0.6519 | 0.0025 | 0.7360 | 0.0036 |
| | PiSA-SR [24] | - | - | - | 0.8283 | 0.0090 | 72.39 | 15.7545 | 0.6657 | 0.0029 | 0.7405 | 0.0023 |
| | PURE [31] | - | - | - | 0.8065 | 0.0087 | 73.31 | 21.4125 | 0.6925 | 0.0028 | 0.7164 | 0.0048 |
| | VARSR [20] | - | - | - | 0.7709 | 0.0065 | 73.82 | 17.9560 | 0.6589 | 0.0031 | 0.7238 | 0.0026 |
| | **NSARM (ours)** | - | - | - | 0.8461 | 0.0033 | 72.85 | 24.3400 | 0.6623 | 0.0034 | 0.7517 | 0.0023 |
| | NSARM *(w/o refiner)* | - | - | - | 0.8662 | 0.0026 | 74.31 | 14.1161 | 0.6931 | 0.0025 | 0.7715 | 0.0023 |

VARSR [20]. Among them, SeeSR, OSEDiff and PiSA-SR are built upon SD 2.1 [23] and SUPIR is built upon SDXL [19], while all of them could generate $1024 \times 1024$ images. SeeSR and SUPIR generate HR from random Gaussian noise while OSEDiff and PiSA-SR generate HR from LR inputs. PURE and VARSR are AR-based methods (VARSR also adopts diffusion-based refinement), which only support generating $512 \times 512$ images.

**Testing Protocol.** Since NSARM is trained for $1024 \times 1024$ resolution while some AR methods (VARSR and PURE) only support $512 \times 512$ resolution, and the commonly used real-world test images are also of resolution $512 \times 512$, we conduct evaluations at both resolutions to ensure a fair comparison. For $1024 \times 1024$ resolution, we collect 100 images cropped from DIV2K Validation [1] and degrade them using the RealESRGAN degradation pipeline to synthesize the LR input. For $512 \times 512$ resolution, following previous works, we adopt the RealSR [2] and DrealSR [32] datasets, which contain LR-GT pairs, and the RP60 [37] dataset, which only has LR images without GT. For methods that can generate $1024 \times 1024$ resolution images, we first generate the HR images and then downsample (using bicubic downsampling) them to $512 \times 512$ for comparison. For methods SeeSR, OSEDiff and PiSA-SR, which can generate $1024 \times 1024$ resolution images but are trained on $512 \times 512$ resolution, we also provide their direct $512 \times 512$ Real-ISR results in the **Appendix C**.

**Evaluation Protocol.** We comprehensively evaluate the competing methods from different aspects, including *output quality*, *model complexity*, and *model robustness*. First, following previous work, we assess the model performance using full-reference metrics (PSNR and SSIM computed on the YCbCr space's Y channel; LPIPS [41] in RGB space) and no-reference metrics (CLIPIQA [27], MUSIQ [13], MANIQA [35], and TOPIQ [3]). We also provide qualitative comparisons to validate the visual effects of the proposed NSARM. Second, we evaluate the model complexity in terms of the number of parameters and inference time. Third and more importantly, we evaluate the model robustness by using four human perception metrics (CLIPIQA, MUSIQ, MANIQA and TOPIQ), which can reflect the stability of models in practical use.




| Method | Deficient | Poor | Collapse |
|---|---|---|---|
| SeeSR | 32 | 10 | 2 |
| SUPIR | 62 | 19 | 13 |
| OSEDiff | 67 | 17 | 4 |
| PiSA-SR | 37 | 8 | 3 |
| NSARM (Ours) | 15 | 1 | 0 |

| Method | Deficient ↓ | Poor ↓ | Collapse ↓ |
|---|---|---|---|
| SeeSR | 27 | 9 | 3 |
| SUPIR | 21 | 5 | 1 |
| OSEDiff | 24 | 3 | 0 |
| PiSA-SR | 21 | 3 | 0 |
| PURE | 24 | 3 | 1 |
| VARSR | 32 | 5 | 0 |
| NSARM (Ours) | 17 | 3 | 0 |

Figure 5: Model robustness performance. The curves show sorted average scores over CLIPIQA, MUSIQ (divided by 100), MANIQA and TOPIQ for different methods on the DIV2K and RP60 datasets. The table lists the numbers of Deficient, Poor and Collapse cases for each method.

We measure the robustness from two aspects. The first is the variance of the perception metrics over each test dataset, where a smaller variance indicates a more robust performance. Second, we sort the perception metric scores from high to low, and count the number of failure cases (*i.e.*, long-tail cases) to evaluate the model robustness. Specifically, we calculate the global mean $\mu_G$ for a metric on a dataset across all methods, and count the number of failure cases at three different levels:

$$\text{Deficient / Poor / Collapse} : \sum_{i=1}^{N} 1[x_i < (1 \,/\, 0.9 \,/\, 0.8) * \mu_G], \tag{8}$$

where Deficient, Poor and Collapse mean the relative image quality. These thresholds are heuristic, but they offer an intuitive characterization of failure severity, as there is currently no standard robustness metric for Real-ISR. Clearly, the model with better robustness should have fewer failure cases.

### 4.2 Comparison with State-of-the-Arts

**Quantitative Quality.** Tab. 1 compares our NSARM with other SD-based and AR-based methods. First, SD-based methods (PiSA-SR, OSEDiff, and SeeSR) and the AR–diffusion hybrid VARSR generally perform well on full-reference fidelity metrics (PSNR/SSIM/LPIPS), benefiting from their continuous VAE representations. Nevertheless, the final NSARM output achieves competitive fidelity and substantially outperforms the pure AR method PURE [31]. Second, for no-reference perceptual metrics (CLIPIQA, MUSIQ, MANIQA and TOPIQ), NSARM obtains overall better results than its competitors under this evaluation protocol, because it tends to generate richer, more natural details favored by human perception. Notably, we also report NSARM *(w/o refiner)*, the intermediate AR output obtained during the same inference run. It ranks first on nearly all no-reference metrics while achieving fidelity scores comparable to those of SUPIR and PURE, making it an attractive option when perceptual quality is prioritized over pixel-level fidelity. VARSR also performs well on no-reference metrics, but it requires an additional diffusion-based refinement process, making its Real-ISR pipeline more complex. Moreover, current IQA metrics cannot fully capture perceptual quality; as shown in Fig. 6 and **Appendix C**, VARSR tends to produce smoother results.

**Robustness.** Beyond the best results in human perception metrics, NSARM demonstrates strong robustness to input images. As can be seen in Tab. 1, NSARM achieves the lowest variances of many perception metrics, indicating that it could perform stably for LR inputs of varying contents and degradations. To more clearly demonstrate the robustness of NSARM, in Fig. 5 we plot the curves of sorted average scores over CLIPIQA, MUSIQ (divided by 100), MANIQA and TOPIQ for different methods on the DIV2K and RP60 datasets. We see that the curve of NSARM lies overall above the other methods, and it drops slowly at the end portion of the curve. In comparison, other methods drop very sharply, which implies that they will generate images of poor visual quality. The table in each sub-figure of Fig. 5 lists the numbers of Deficient, Poor and Collapse cases (refer to Eq. 8) for each method. It can be observed that our model yields the fewest Deficient and Poor cases across all datasets, with no Collapse examples. SeeSR and SUPIR exhibit the most frequent Collapse failures because they generate HR images from Gaussian noise, which introduces more randomness. VARSR shows improved robustness through full fine-tuning (similar to our approach), yet remains limited by its non-LR starting point and discrete tokenization bottleneck, underperforming NSARM in both robustness and overall quality. This comprehensive validation confirms that NSARM is more

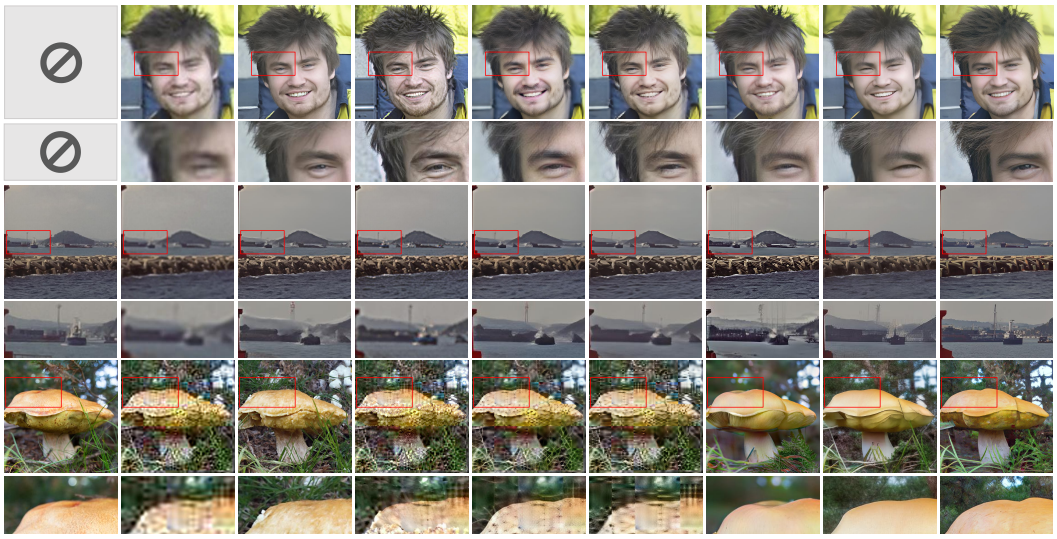



Figure 6: Visual comparison on RP60, RealSR and DIV2K datasets (**zoom in for a better view**).

| Method | Inference Time (s) | Total Param (M) |
|---|---|---|
| VARSR [20] | 0.3 (512) | 1,102 |
| PURE [31] | 380 (512) | 7,080 |
| OSEDiff [33] | 0.4 (1024) | 1,300 |
| PiSA-SR [24] | 0.4 (1024) | 1,775 |
| SeeSR [34] | 13 (1024) | 2,524 |
| SUPIR [37] | 11.7 (1024) | 4,801 |
| NSARM (Ours) | 1.3 (1024) | 8,770 |

Table 2: Complexity comparison of different methods. PURE and VARSR can only generate $512 \times 512$ images, while the others are evaluated on $1024 \times 1024$ resolution. The inference time is measured on an A800 80G GPU.

robust than the competing methods. More detailed score distributions for each metric and dataset are provided in the **Appendix C**.

**Qualitative Comparisons.** Visual comparisons with different Real-ISR methods are presented in Fig. 6. In the first case, our approach reconstructs the most natural facial features and hair textures with exceptional clarity, while the competing methods exhibit significant shortcomings: SUPIR introduces over-sharp artifacts, and OSEDiff and PiSA-SR produce noticeable ghosting effects. The second case demonstrates our method's superior detail recovery capability: NSARM achieves photorealistic restoration and can handle the input with very low quality. The third example demonstrates the importance of model robustness: with challenging degradations, many methods (particularly those based on SD) will collapse, while NSARM maintains robust performance for those challenging images. Actually, due to the limitations of current IQA metrics for generative ISR, numerically similar scores may mask the substantial perceptual differences. In fact, visual effects of NSARM are much better than other methods. Substantially more evidence is provided in the **Appendix C and D**.

**Complexity.** Tab. 2 compares the computational complexity of competing Real-ISR methods, including the number of parameters and inference time measured on a single A800 GPU. Note that the inference time of VARSR and PURE is measured at $512 \times 512$ resolution, while others are measured at $1024 \times 1024$ resolution. NSARM demonstrates substantial inference speed advantages over iterative diffusion models, approximately $10\times$ faster than SUPIR and SeeSR, while maintaining competitive speed with single-step methods, trailing OSEDiff and PiSA-SR by less than one second. Notably, our approach achieves over 100x acceleration compared to the pure AR method PURE, while roughly matching the efficiency of VARSR (reported 0.3s is for $512 \times 512$ generation). In terms of parameter scale, NSARM is not advantageous because of its foundation model backbone. Despite the large number of parameters, NSARM predicts next-scale tokens in bitwise space combined with infinite-vocabulary classifier technology, which enables efficient training and inference.




Figure 7: Qualitative examples of different training settings and control methods.

Table 3: Ablation on the two-stage design and on the conditioning mechanism, evaluated on RealSR.

| Method (RealSR) | PSNR↑ | MUSIQ↑ | Training time | Inference time | Batch size | Iterations |
|---|---|---|---|---|---|---|
| Infinity + ControlNet (cross-attn) | 16.85 | 60.67 | 122 h | 2.8 s | 32 | 40K |
| Infinity + Prefix tokens | 17.56 | 62.30 | 116 h | 2.1 s | 32 | 40K |
| NSARM, Stage 1 only | 20.04 | 65.88 | 9 h | **1.3 s** | 196 | 100K |
| NSARM, Stage 2 only (w/o Stage 1) | 15.92 | 61.22 | 92 h | **1.3 s** | 32 | 40K |
| NSARM (two-stage, w/o refiner) | **23.52** | **71.83** | 9 h + 92 h | **1.3 s** | 196 / 32 | 100K / 40K |

### 4.3 Ablation Study

**Pathway Control with Two-Stage Optimization.** Fig. 7 and Tab. 3 demonstrate the necessity of pathway control and two-stage optimization. All Stage 2 variants in this ablation are trained and evaluated without the fidelity refiner. Stage 1 Only preserves the LR structure but produces blur and artifacts because the transformation network cannot fully align with the subsequent AR process. In contrast, Stage 2 Only, cross-attention control from PASD [36], and prefix control from VARSR [20] all yield extremely low PSNR values, indicating a severe loss of input fidelity. Although their MUSIQ scores appear reasonable, the visual results in Fig. 7 reveal that these variants fail to reconstruct the input content; their seemingly acceptable scores mainly arise from globally sharp-looking outputs rather than faithful restoration. Their common limitation is that approximately 100 hours of training provides only 40K iterations at batch size 32, which is insufficient for an 8B AR model to learn an out-of-distribution conditioning pathway from scratch at $1024^2$ resolution. Stage 1 addresses this issue efficiently: the lightweight transformation network is trained in 9 hours for 100K iterations with a batch size of 196, providing an approximately aligned pathway that Stage 2 can refine under the original pre-training objective. This design is also structurally efficient because it operates only on LR scales, whereas cross-attention must be computed at every scale up to $1024^2$, resulting in approximately $2.2\times$ slower inference.

Due to the limited space, we provide more ablation studies in the **Appendix B**.

## 5 Conclusion

We presented NSARM, a robust and efficient framework for Real-ISR tasks, which effectively integrated the generative prior of the pre-trained Infinity model with a novel pathway control mechanism. Using bitwise next-scale prediction, NSARM achieved efficient, high-quality reconstruction through progressive refinement from LR inputs to HR outputs. Our two-stage training strategy, replacing preliminary scales by LR via a transformation network followed by full-parameter fine-tuning of AR backbone and the optimization of a fidelity refiner, ensured exceptional robustness while preserving strong generative capabilities. Extensive experiments demonstrated that NSARM achieves competitive and often superior perceptual quality with very few collapse cases across diverse scenarios, outperforming existing methods in both perceptual metrics and model robustness. Our work not only established a new framework for Real-ISR, but also validated that pure AR models can be used to build promising paradigms for low-level vision tasks.

**Limitations.** As an early exploration of pure AR modeling for Real-ISR, its model size is relatively large. However, it is comparable to recent AR approaches such as PURE, while achieving much faster inference speed. The model size can be compressed by techniques such as pruning.

## References


[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017.

[2] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3086–3095, 2019.

[3] Chaofeng Chen, Jiadi Mo, Jingwen Hou, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Topiq: A top-down approach from semantics to distortions for image quality assessment. *IEEE Transactions on Image Processing*, 33:2404–2418, 2024.

[4] Xiangyu Chen, Xintao Wang, Wenlong Zhang, Xiangtao Kong, Yu Qiao, Jiantao Zhou, and Chao Dong. Hat: Hybrid attention transformer for image restoration. *arXiv preprint arXiv:2309.05239*, 2023.

[5] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11065–11074, 2019.

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 184–199. Springer, 2014.

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

[9] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[11] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. *arXiv preprint arXiv:2412.04431*, 2024.

[12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[13] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5148–5157, 2021.

[14] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[15] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Ben Fei, Bo Dai, Wanli Ouyang, Yu Qiao, and Chao Dong. Diffbir: Towards blind image restoration with generative diffusion prior. *arXiv preprint arXiv:2308.15070*, 2023.

[16] Dongyang Liu, Shitian Zhao, Le Zhuo, Weifeng Lin, Yi Xin, Xinyue Li, Qi Qin, Yu Qiao, Hongsheng Li, and Peng Gao. Lumina-mgpt: Illuminate flexible photorealistic text-to-image generation with multimodal generative pretraining. *arXiv preprint arXiv:2408.02657*, 2024.

[17] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024.

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[19] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[20] Yunpeng Qu, Kun Yuan, Jinhua Hao, Kai Zhao, Qizhi Xie, Ming Sun, and Chao Zhou. Visual autoregressive modeling for image super-resolution. *arXiv preprint arXiv:2501.18993*, 2025.

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *arXiv e-prints*, page arXiv:2103.00020, February 2021.

[22] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[24] Lingchen Sun, Rongyuan Wu, Zhiyuan Ma, Shuaizheng Liu, Qiaosi Yi, and Lei Zhang. Pixel-level and semantic-level adjustable super-resolution: A dual-lora approach. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2333–2343, 2025.

[25] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024.

[26] Unsplash-website. Unsplash-website. `https://unsplash.com/data`.

[27] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2555–2563, 2023.

[28] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *International Journal of Computer Vision*, 132(12):5929–5949, 2024.

[29] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1905–1914, 2021.

[30] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[31] Hongyang Wei, Shuaizheng Liu, Chun Yuan, and Lei Zhang. Perceive, understand and restore: Real-world image super-resolution with autoregressive multimodal generative models. *arXiv preprint arXiv:2503.11073*, 2025.

[32] Pengxu Wei, Ziwei Xie, Hannan Lu, Zongyuan Zhan, Qixiang Ye, Wangmeng Zuo, and Liang Lin. Component divide-and-conquer for real-world image super-resolution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 101–117. Springer, 2020.

[33] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *Advances in Neural Information Processing Systems*, 37:92529–92553, 2024.

[34] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25456–25467, 2024.

[35] Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1191–1200, 2022.

[36] Tao Yang, Rongyuan Wu, Peiran Ren, Xuansong Xie, and Lei Zhang. Pixel-aware stable diffusion for realistic image super-resolution and personalized stylization. In *European conference on computer vision*, pages 74–91. Springer, 2024.

[37] Fanghua Yu, Jinjin Gu, Zheyuan Li, Jinfan Hu, Xiangtao Kong, Xintao Wang, Jingwen He, Yu Qiao, and Chao Dong. Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25669–25680, 2024.

[38] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

[39] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *arXiv preprint arXiv:2307.12348*, 2023.

[40] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4791–4800, 2021.

[41] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[42] Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3096–3105, 2019.

[43] Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary spherical quantization. *arXiv preprint arXiv:2406.07548*, 2024.

# Appendix

In this appendix, we provide the following materials:

**A.** Pseudo code of NSARM, including the visual tokenizer encoding process and cascaded GT modification used in training (referring to Sec. 3).

**B.** Ablation study and user study, including analyses of the replaced scale $k$ and text prompts (referring to Sec. 3.2 and Sec. 4).

**C.** More experimental results, including additional visual comparisons, direct 512×512 evaluation results, and detailed sorted score distributions (referring to Sec. 4).

**D.** Discussions on collapsed and worst cases, including robustness under different degradation levels and challenging real-world examples (referring to Sec. 4).

## A Pseudo Code of NSARM

To facilitate understanding, we provide the pseudo code of NSARM. Alg. A.1 shows Visual Tokenizer Encoding, which obtains residuals from GT. The GT feature $\boldsymbol{F}$ is decomposed into the residual queue $\boldsymbol{R}_{queue}$ and accumulated features $\widetilde{\boldsymbol{F}}_{queue}$. In Infinity, $\widetilde{\boldsymbol{F}}_{queue}$ provides the inputs of each scale, while $\boldsymbol{R}_{queue}$ provides the corresponding residual labels. Our transformation network $\mathcal{T}(\cdot)$ maps LR to the preliminary scale residuals (the first seven elements of $\boldsymbol{R}_{queue}$), which are optimized in Stage 1.

We then obtain transformed residuals $\boldsymbol{R}'_{queue} = \{\mathcal{T}(LR)_1, \mathcal{T}(LR)_2, \ldots, \mathcal{T}(LR)_7\}$ to replace the original preliminary scales. During Stage 2, NSARM uses modified GT residual labels to constrain the generation pathway from $\boldsymbol{R}'_{queue}$ to GT. As shown in Alg. A.2, replacing the preliminary residuals causes a cascaded update of $\boldsymbol{R}_{queue_m}$ and $\widetilde{\boldsymbol{F}}_{queue_m}$. Starting from $\boldsymbol{R}'_{queue}$, each subsequent step is refined by the GT feature $\boldsymbol{F}$. This yields the supervision signal $\boldsymbol{R}_{queue_m}$ for $\mathcal{L}_{AR}$ in Stage 2. The fidelity refiner further adjusts $\boldsymbol{F}_k$ with $\mathcal{L}_{MSE}$.

**Algorithm A.1** Visual Tokenizer Encoding

**Input**: Raw feature $\boldsymbol{F}$,
scale schedule $\{(h_1^r, w_1^r), ..., (h_K^r, w_K^r)\}$
**Output**: $\boldsymbol{R}_{queue}, \widetilde{\boldsymbol{F}}_{queue}$

$\boldsymbol{R}_{queue} = []$ (multi-scale bit labels)
$\widetilde{\boldsymbol{F}}_{queue} = []$ (inputs for AR, accumulated $\boldsymbol{R}$)
**for** $k = 1, 2, \cdots, K$ **do**
  $\boldsymbol{R}_k = \mathcal{Q}(\mathrm{down}(\boldsymbol{F} - \boldsymbol{F}_{k-1}, (h_k, w_k)))$
  $\mathrm{Queue_Push}(\boldsymbol{R}_{queue}, \boldsymbol{R}_k)$
  **if** $k < K$ **then**
    $\boldsymbol{F}_k = \sum_{i=1}^{k} \mathrm{up}(\boldsymbol{R}_i, (h, w))$
    $\widetilde{\boldsymbol{F}}_k = \mathrm{down}(\boldsymbol{F}_k, (h_{k+1}, w_{k+1}))$
    $\mathrm{Queue_Push}(\widetilde{\boldsymbol{F}}_{queue}, \widetilde{\boldsymbol{F}}_k)$
  **end if**
**end for**

## B Ablation Study and User Study

### B.1 Ablation Study

**Impact of replaced scale $k$.** As explained in the main paper, $k = 7$ corresponds to 256x256, which retains LQ information (no down-sampling) while delegating the generation to the AR model. When $k$ is small, fidelity drops (see $k = 2$, 4 in Fig. 3 of the main paper), because the input resolution is smaller and the generation process becomes closer to pure T2I generation. When $k$ is large (see

**Algorithm A.2** Cascaded GT Modification

**Input**: Raw feature $\boldsymbol{F}$,
scale schedule $\{(h_1^r, w_1^r), ..., (h_K^r, w_K^r)\}$,
transformed LR $\boldsymbol{R}'_{queue}$

**Output**: $\boldsymbol{R}_{queue_m}, \widetilde{\boldsymbol{F}}_{queue_m}, \boldsymbol{F}_k$

$\boldsymbol{R}_{queue_m} = []$ (modified multi-scale bit labels)
$\widetilde{\boldsymbol{F}}_{queue_m} = []$ (modified inputs, accumulated $\boldsymbol{R}$)
**for** $k = 1, 2, \cdots, K$ **do**
  **if** $k <= 7$ **then**
    $\boldsymbol{R}_k = \boldsymbol{R}'_{queue}[k]$
  **else**
    $\boldsymbol{R}_k = \mathcal{Q}(\text{down}(\boldsymbol{F} - \boldsymbol{F}_{k-1}, (h_k, w_k)))$
  **end if**
  Queue_Push($\boldsymbol{R}_{queue_m}, \boldsymbol{R}_k$)
  **if** $k < K$ **then**
    $\boldsymbol{F}_k = \sum_{i=1}^{k} \text{up}(\boldsymbol{R}_i, (h, w))$
    $\widetilde{\boldsymbol{F}}_k = \text{down}(\boldsymbol{F}_k, (h_{k+1}, w_{k+1}))$
    Queue_Push($\widetilde{\boldsymbol{F}}_{queue_m}, \widetilde{\boldsymbol{F}}_k$)
  **end if**
**end for**

| $k$ scales (Test on RealSR) | 0 or 1 | 5 | 7 | 9 | 13 |
|---|---|---|---|---|---|
| PSNR ↑ | Become a | 21.67 | 24.83 | 25.78 | Become a classical |
| MUSIQ ↑ | T2I model. | 71.32 | 70.45 | 70.35 | SR model with $L_{MSE}$ Loss. |

Table B.1: Performance comparison of NSARM with different $k$.

| Methods on RealSR | PSNR↑ | CLIPIQA↑ | MUSIQ↑ | MANIQA↑ | TOPIQ↑ | Avg. Variance↓ |
|---|---|---|---|---|---|---|
| NSARM (Fixed) | 24.88 | 0.7310 | 70.26 | 0.6058 | 0.7036 | 0.0035 |
| NSARM (LLaVA) | 24.83 | 0.7309 | 70.45 | 0.6067 | 0.7204 | 0.0036 |

Table B.2: Performance comparison of NSARM with different text prompts.

Tab. B.1), fidelity improves but perceptual metrics decline. Therefore, the model exhibits a clear tendency toward generation or restoration under different $k$ values. This aligns with our understanding: if we ignore the fidelity refiner, $k = 1$ would reduce the model to a pure T2I model, whereas $k = 13$ would turn it into a conventional SR model. Moreover, using a larger $k$ significantly increases the computational cost. Therefore, we choose $k = 7$ as a good trade-off between perceptual quality, fidelity, and efficiency.

**Impact of Text Prompts.** As shown in Fig. 2 of the main paper, NSARM accepts text inputs that can be either content descriptions from a multimodal model (*e.g.*, LLaVA [17]) or a fixed default prompt (*e.g.* "A high-quality image with harmonious colors and rich details"). As shown in Tab. B.2, the performance gap between these two variants is marginal: LLaVA-generated prompts yield slightly higher perceptual scores, while the fixed prompt achieves slightly better PSNR. This weak dependence on prompt can be attributed to our scale replacement strategy. In Infinity, the first scale is derived from text, and prompts mainly affect early-scale generations; in NSARM, these early scales are replaced by the transformation network output. Our results suggest that prompt choice has limited impact on NSARM, and simple fixed prompts suffice for most scenarios.

### B.2 User Study

To comprehensively evaluate the perceptual quality in practice, we conducted a user study with 12 participants. From each of the benchmark datasets (DIV2K [1], DRealSR [32], RealSR [2], and RP60 [37]), we randomly selected 10 representative images (40 total) and generated the outputs of the compared methods. Participants performed blind evaluations by selecting the best result from anonymized options (Fig. B.1), with GT provided as the reference when available. Across the 480 total votes (12 participants × 40 images), our method received 272 first-choice selections ( 56.7%), demonstrating clear user preference (Fig. B.1). VARSR ranked second (14% votes), benefiting from its VAR framework. OSEDiff and PiSA-SR (single-step diffusion) performed weaker (8% and 6%




(a) The Results of User Study.

(b) Interface of User Study. Participants are asked to select the best one from 1 to 7.

Figure B.1: The results of user study. The participants are asked to select the best one from the 7 methods (the order of 1-7 was randomly changed in different test cases).

respectively), despite competitive numerical metrics. This reveals that their speed advantage comes at the cost of perceptual quality.

# C More Experimental Results

**More Visual Results.** As mentioned in the section of Qualitative Comparisons in the main paper, current IQA metrics for Real-ISR have limitations, where numerically similar scores may obscure significant perceptual differences. Here, we provide additional visual comparisons to demonstrate NSARM's robustness. Figs. C.2 to C.5 present results on DIV2K [1], DRealSR [32], RealSR [2], and RP60 [37] datasets, respectively. Note that, due to layout constraints and considering that PiSA-SR (as an enhanced version of OSEDiff) produces similar effects to OSEDiff, we only present the result of PiSA-SR for comparison.

In Fig. C.2, we see that our method consistently produces clearer visual results without introducing background artifacts. Unlike PURE or SUPIR, NSARM avoids the generation of over-sharpening or hallucinated details. Fig. C.3 demonstrates the superiority of NSARM in handling smooth areas and distinct details (such as clean, sharp text). It produces little artifacts or flaws in smooth areas, and little illusions or over-sharpening in areas with many details. Fig. C.4 shows that the outputs of NSARM match closely the visual quality of GT: it accurately recovers cherry blossoms and branches in the first row (while SUPIR mistakenly interprets them as snow-covered trees). In the third row, NSARM preserves smooth white areas while correctly rendering tiled roof textures. The first row of Fig. C.5 highlights NSARM's advantages in facial and hair reconstruction. The second row shows that NSARM preserves star details without mistaking them for noise. Besides, NSARM successfully recovers "frozen roses in snowy scenes" (third row) where many methods fail.

Our visual analysis reveals the distinct characteristics among the compared methods. SUPIR and PURE frequently produce over-sharpened results with hallucinated content, particularly in texture-rich regions (*e.g.*, the first and second samples of Fig. C.2 and the first row of Fig. C.4). PiSA-SR, as a single-stage diffusion model, exhibits limited generative capability, often yielding oversmoothed outputs or visible artifacts (*e.g.*, the first sample of both Fig. C.4 and Fig. C.5). Although VARSR achieves competitive numerical metrics (as reported in the main paper), its reconstructions tend to be excessively smooth. In contrast, our NSARM maintains a good balance, avoiding both the hallucination of SUPIR/PURE and the over-smoothing issues of PiSA-SR/VARSR, while consistently producing the most faithful reconstructions across most test cases.

**More Quantitative Results.** As mentioned in the Testing Protocol section of the main paper, while SeeSR, OSEDiff, and PiSA-SR are originally trained on $512 \times 512$ images, we compare with them by first generating $1024 \times 1024$ outputs followed by downsampling to $512 \times 512$ for unified evaluation. In Tab. C.3, we additionally present results from these methods when directly generating $512 \times 512$ images. The results reveal two observations: (1) The "first generating $1024 \times 1024$ then downsampling to $512 \times 512$" approach generally achieves better fidelity metrics, benefiting from the additional computational resources. (2) For perceptual metrics, performance varies, with some metrics improving while others degrade. Importantly, even when comparing our downsampled

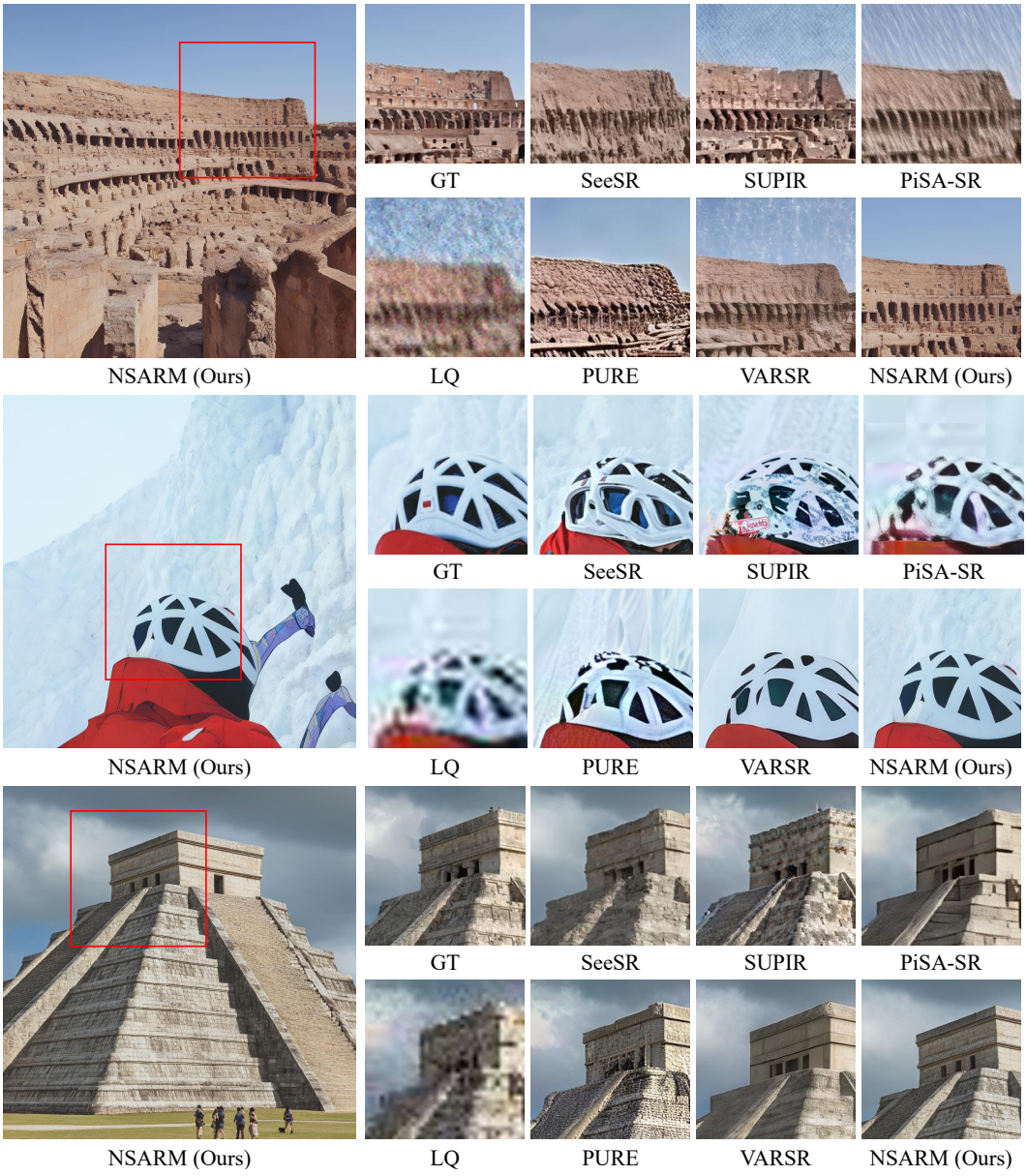



Figure C.2: Visual comparison of different methods on DIV2K (zoom in for a better view).

$512 \times 512$ results with natively trained models, NSARM maintains superior performance on most perceptual quality measures, demonstrating its robustness.

**More Sorted Score Distributions.** In the main paper, we presented two representative curves showing sorted average scores across CLIP-IQA, MUSIQ (divided by 100), MANIQA and TOPIQ metrics for different methods on DIV2K and RP60 datasets. For a more comprehensive analysis, we provide the per-metric and per-dataset results in Figs. C.6 and C.7 (16 curves in total). These extended results demonstrate that: (1) the curves of NSARM are higher than those of other approaches in most cases, maintaining superior positions throughout the score distributions; and (2) the gradual descent of NSARM's curves, which indicates fewer bad quality outliers, confirms the robustness of NSARM. This consistent performance further validates NSARM's superior reconstruction quality and exceptional robustness.

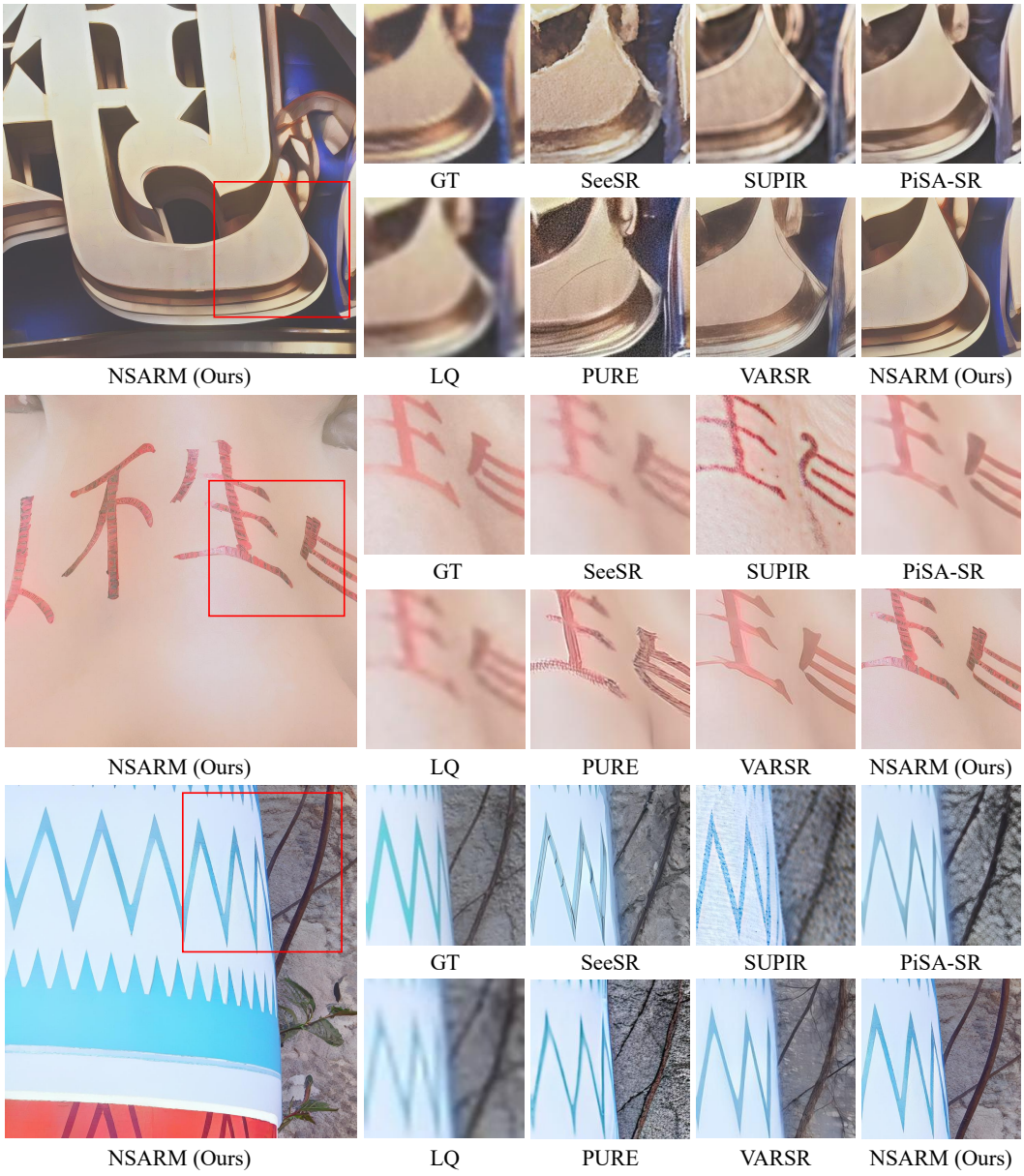



Figure C.3: Visual comparison of different methods on DRealSR (zoom in for a better view).

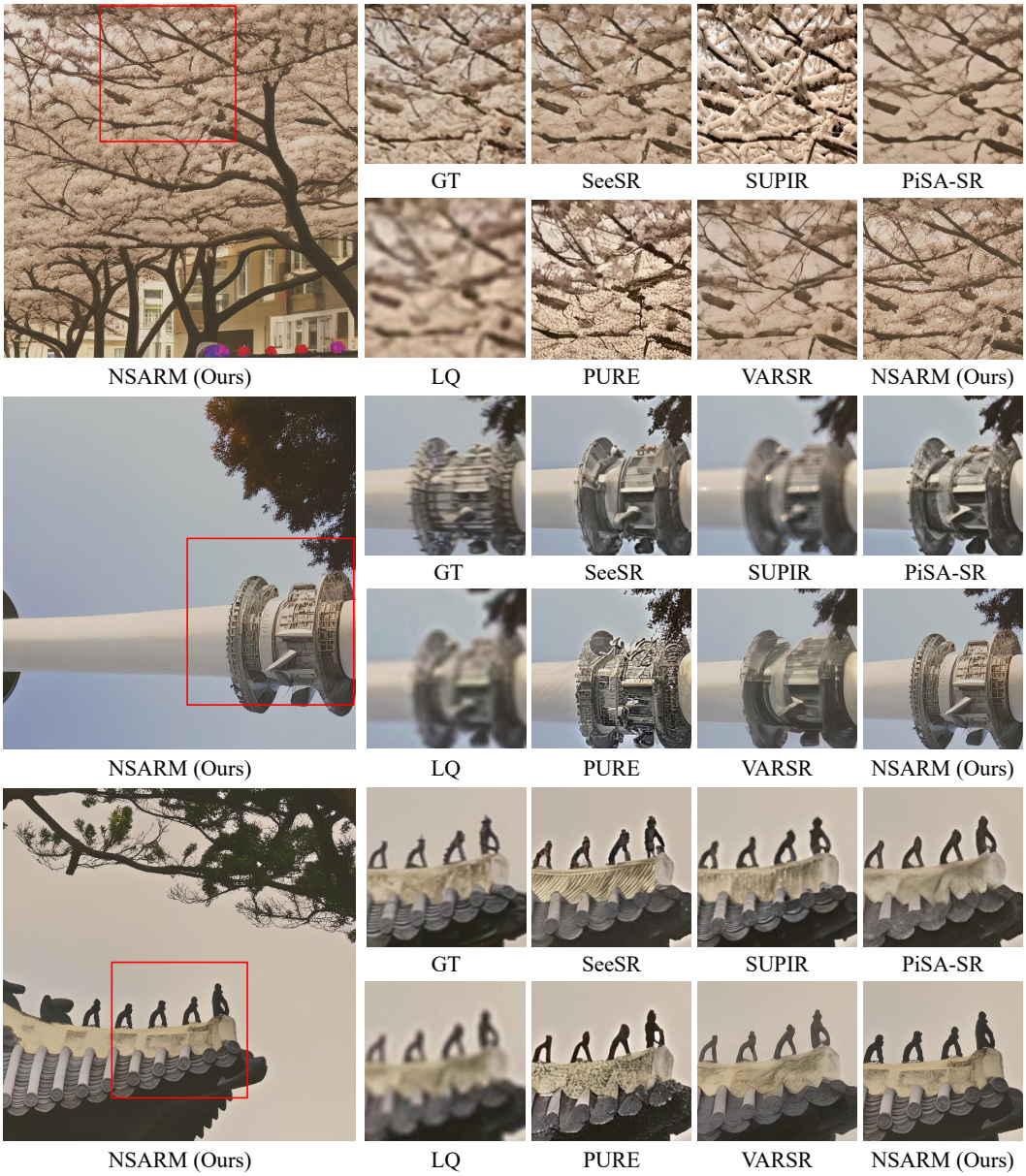



Figure C.4: Visual comparison of different methods on RealSR (zoom in for a better view).

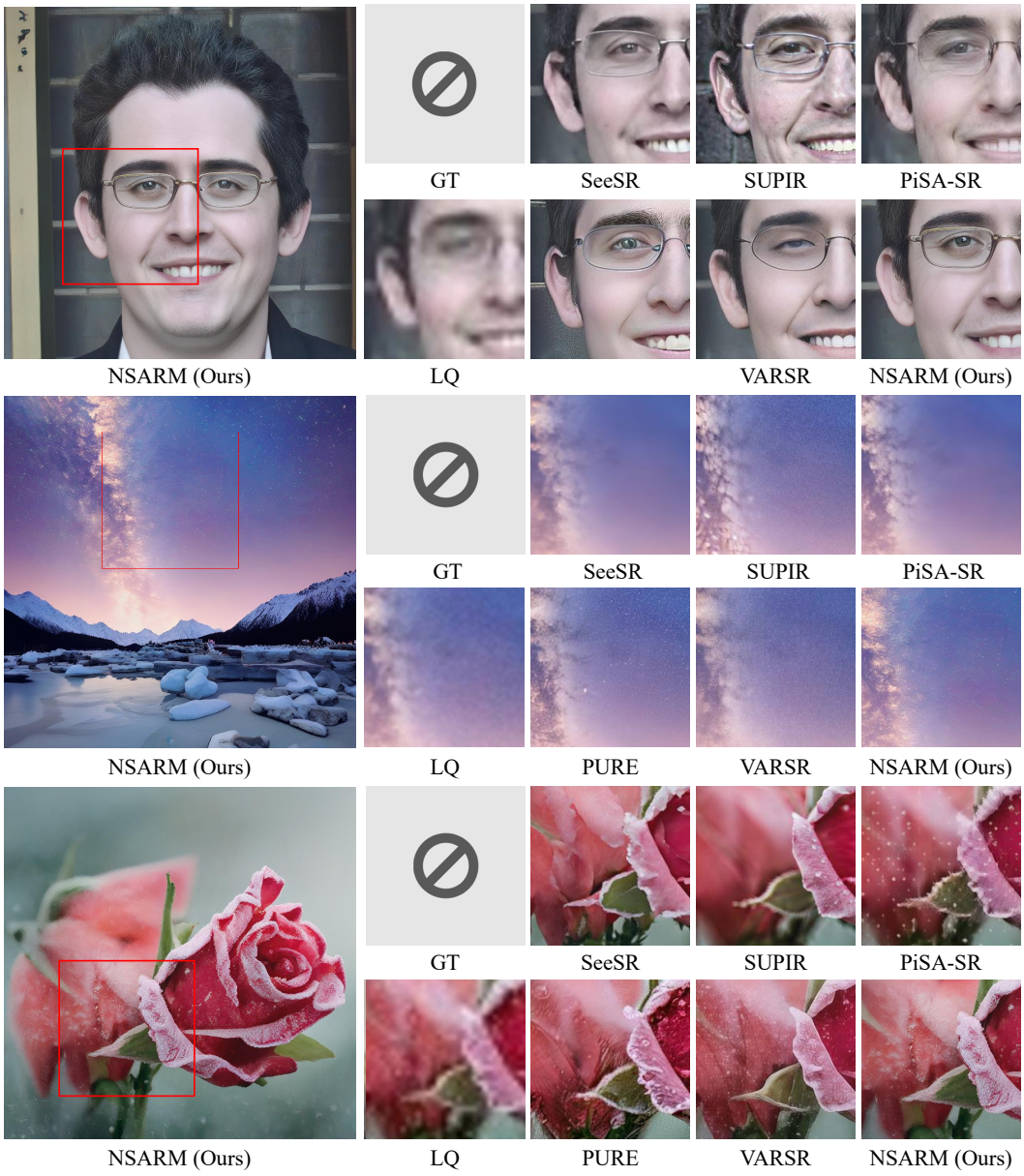



Figure C.5: Visual comparison of different methods on RP60 (zoom in for a better view).

| Datasets | Methods | PSNR↑ | SSIM↑ | LPIPS↓ | CLIPIQA | | MUSIQ | | MANIQA | | TOPIQ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Avg.↑ | Var.↓ | Avg.↑ | Var.↓ | Avg.↑ | Var.↓ | Avg.↑ | Var.↓ |
| DRealSR | SeeSR [34] | 28.76 | 0.7762 | 0.3316 | 0.6401 | 0.0217 | 61.11 | 174.4791 | 0.5603 | 0.0123 | 0.6217 | 0.0207 |
| | SeeSR (512) | 28.07 | 0.7684 | 0.3174 | 0.6908 | 0.0152 | 65.09 | 111.4403 | 0.6042 | 0.0068 | 0.6575 | 0.0111 |
| | OSEDiff [33] | 28.60 | 0.7872 | 0.3296 | 0.7082 | 0.0161 | 60.91 | 142.5506 | 0.5350 | 0.0089 | 0.6208 | 0.0113 |
| | OSEDiff (512) | 27.92 | 0.7836 | 0.2967 | 0.6952 | 0.0136 | 64.68 | 86.7438 | 0.5895 | 0.0038 | 0.6002 | 0.0051 |
| | PiSA-SR [24] | 28.88 | 0.7974 | 0.3145 | 0.6453 | 0.0217 | 60.33 | 131.9325 | 0.5243 | 0.0091 | 0.5987 | 0.0130 |
| | PiSA-SR (512) | 28.32 | 0.7804 | 0.2960 | 0.6967 | 0.0123 | 66.11 | 92.9261 | 0.6153 | 0.0046 | 0.6334 | 0.0060 |
| | NSARM (Ours) | 28.05 | 0.7381 | 0.3919 | 0.7231 | 0.0077 | 66.68 | 92.7997 | 0.5759 | 0.0057 | 0.6933 | 0.0062 |
| RealSR | SeeSR [34] | 25.52 | 0.7369 | 0.3065 | 0.6663 | 0.0141 | 68.57 | 44.2620 | 0.6222 | 0.0052 | 0.6908 | 0.0063 |
| | SeeSR (512) | 25.42 | 0.7295 | 0.2984 | 0.6537 | 0.0134 | 69.25 | 37.6640 | 0.6396 | 0.0041 | 0.6795 | 0.0076 |
| | OSEDiff [33] | 25.52 | 0.7334 | 0.3275 | 0.7393 | 0.0081 | 68.65 | 34.3655 | 0.6265 | 0.0037 | 0.6924 | 0.0035 |
| | OSEDiff (512) | 25.15 | 0.7341 | 0.2920 | 0.6690 | 0.0126 | 69.10 | 25.9077 | 0.6330 | 0.0023 | 0.6253 | 0.0039 |
| | PiSA-SR [24] | 25.91 | 0.7526 | 0.2902 | 0.6803 | 0.0104 | 69.14 | 29.6003 | 0.6259 | 0.0027 | 0.6885 | 0.0029 |
| | PiSA-SR (512) | 25.50 | 0.7418 | 0.2672 | 0.6699 | 0.0111 | 70.15 | 26.3891 | 0.6552 | 0.0024 | 0.6373 | 0.0044 |
| | NSARM (Ours) | 24.83 | 0.6758 | 0.3655 | 0.7309 | 0.0054 | 70.45 | 27.6000 | 0.6067 | 0.0038 | 0.7204 | 0.0023 |
| RP60 | SeeSR [34] | - | - | - | 0.7881 | 0.0184 | 71.86 | 21.8667 | 0.6654 | 0.0023 | 0.7253 | 0.0061 |
| | SeeSR (512) | - | - | - | 0.7654 | 0.0128 | 72.20 | 23.4764 | 0.6649 | 0.0020 | 0.7268 | 0.0056 |
| | OSEDiff [33] | - | - | - | 0.8424 | 0.0070 | 71.88 | 17.1801 | 0.6519 | 0.0025 | 0.7360 | 0.0036 |
| | OSEDiff (512) | - | - | - | 0.7480 | 0.0073 | 71.29 | 22.1331 | 0.6551 | 0.0017 | 0.6525 | 0.0042 |
| | PiSA-SR [24] | - | - | - | 0.8283 | 0.0090 | 72.39 | 15.7545 | 0.6657 | 0.0029 | 0.7405 | 0.0023 |
| | PiSA-SR (512) | - | - | - | 0.7800 | 0.0072 | 73.50 | 13.6328 | 0.6889 | 0.0018 | 0.7068 | 0.0029 |
| | NSARM (Ours) | - | - | - | 0.8461 | 0.0033 | 72.85 | 24.3400 | 0.6623 | 0.0034 | 0.7517 | 0.0023 |

Table C.3: Performance comparison of different Real-ISR methods on commonly used datasets. "(512)" means that the method directly generates $512\times512$ results. Red indicates the best performance. $\downarrow$ and $\uparrow$ indicate that lower and higher values are better, respectively.

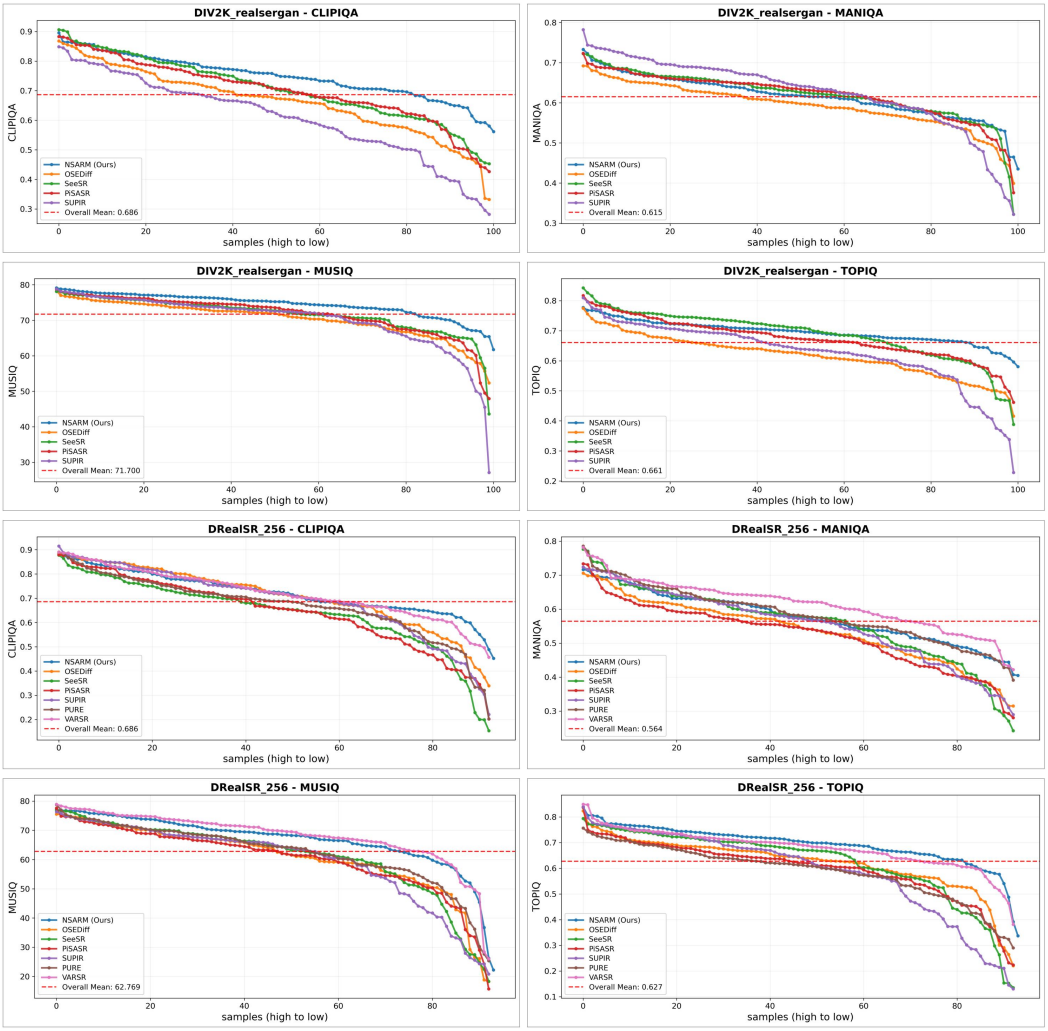



Figure C.6: Detailed ranked score distributions of Real-ISR methods across multiple evaluation metrics on DIV2K and DRealSR.

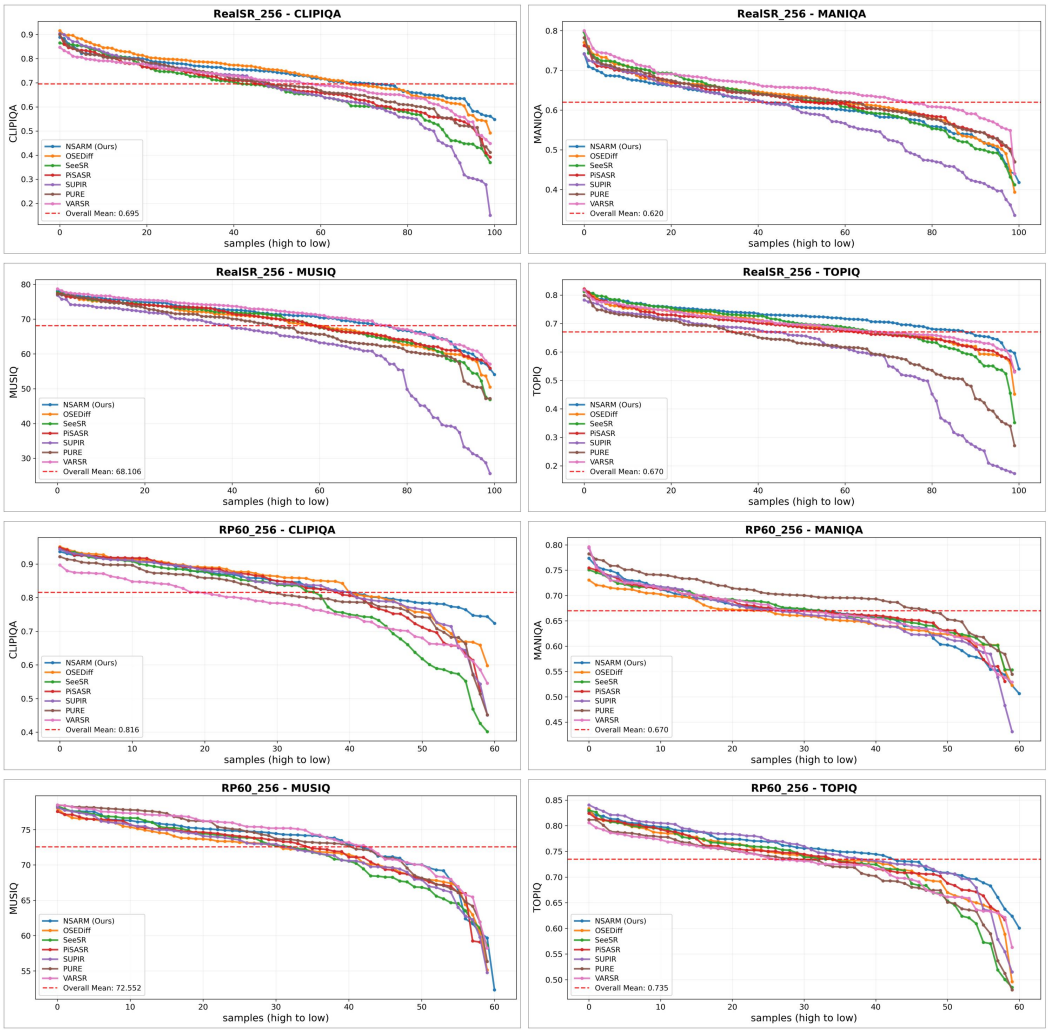



Figure C.7: Detailed ranked score distributions of Real-ISR methods across multiple evaluation metrics on RealSR and RP60.

# D Discussions on Collapsed and Worst Cases

**Collapsed Cases.** We first investigate how different degradation strengths affect model performance and when outputs begin to "collapse". We check the training degradation settings of each method: NSARM and SeeSR both use the original RealESRGAN degradation pattern and strength during training; PiSA-SR and VARSR also use the RealESRGAN pipeline but adopt a weaker degradation strength range; SUPIR does not disclose its detailed degradation settings. Based on these ranges, we define seven degradation levels: Levels 1–5 lie within the training distribution of all methods; Level 6 corresponds to the maximum degradation strength in the original RealESRGAN design (and thus the upper bound for the training of SeeSR and NSARM); Level 7 represents a heavier degradation strength beyond the training settings of all methods.

As shown in Fig. D.8, PiSA-SR starts to collapse from Level 4, producing noticeably blurred results, even though this level remains within its training range. At Levels 5 and 6, SeeSR shows clear collapse, with artifacts such as the lion's tongue appearing furry. SUPIR tends to over-sharpen and loses fine fur texture above Level 5. VARSR consistently produces blurry results and even removes the lion's whiskers from Level 5 onward. In contrast, our method exhibits strong robustness and preserves superior visual quality at all levels, even at Level 7, which lies outside the training distribution.

The results in Fig. D.9 are largely consistent with Fig. D.8. However, PiSA-SR at Level 4 and VARSR at Level 5 recover the dark fence structure reasonably well in Fig. D.9. This indicates that model collapse depends on both the image content and the degradation strength. Overall, these results demonstrate that NSARM offers clear advantages in both robustness and visual quality.

**Worst Cases on Real-World Datasets.** In the metric curves presented in the main paper and Figs. C.6 and C.7 of this supplementary file, we rank the scores of various metrics of each method to evaluate the model robustness. Although these curves visualize the distribution of metric scores across different samples, the comparison at each position of the abscissa may not correspond to the same input image. Therefore, in this section, we show in Figs. D.10 and D.11 the five images with the lowest perceptual metric scores (the average scores over CLIPIQA, MUSIQ divided by 100, MANIQA, and TOPIQ) for each method on the RealSR and RP60 datasets, respectively, revealing the specific images that receive poor scores. We can see that the images with the worst scores largely overlap among methods, indicating that our approach genuinely improves the performance on challenging images where many methods fail. In Fig. D.12, we show the images on which NSARM performs better than other methods, sorted by the improvement between NSARM's scores and the average scores over competing methods. Notably, many of these images coincide with the worst-case images shown in Figs. D.10 and D.11. This consistency further validates that our method achieves better performance on those low-scoring images, demonstrating its robustness performance.

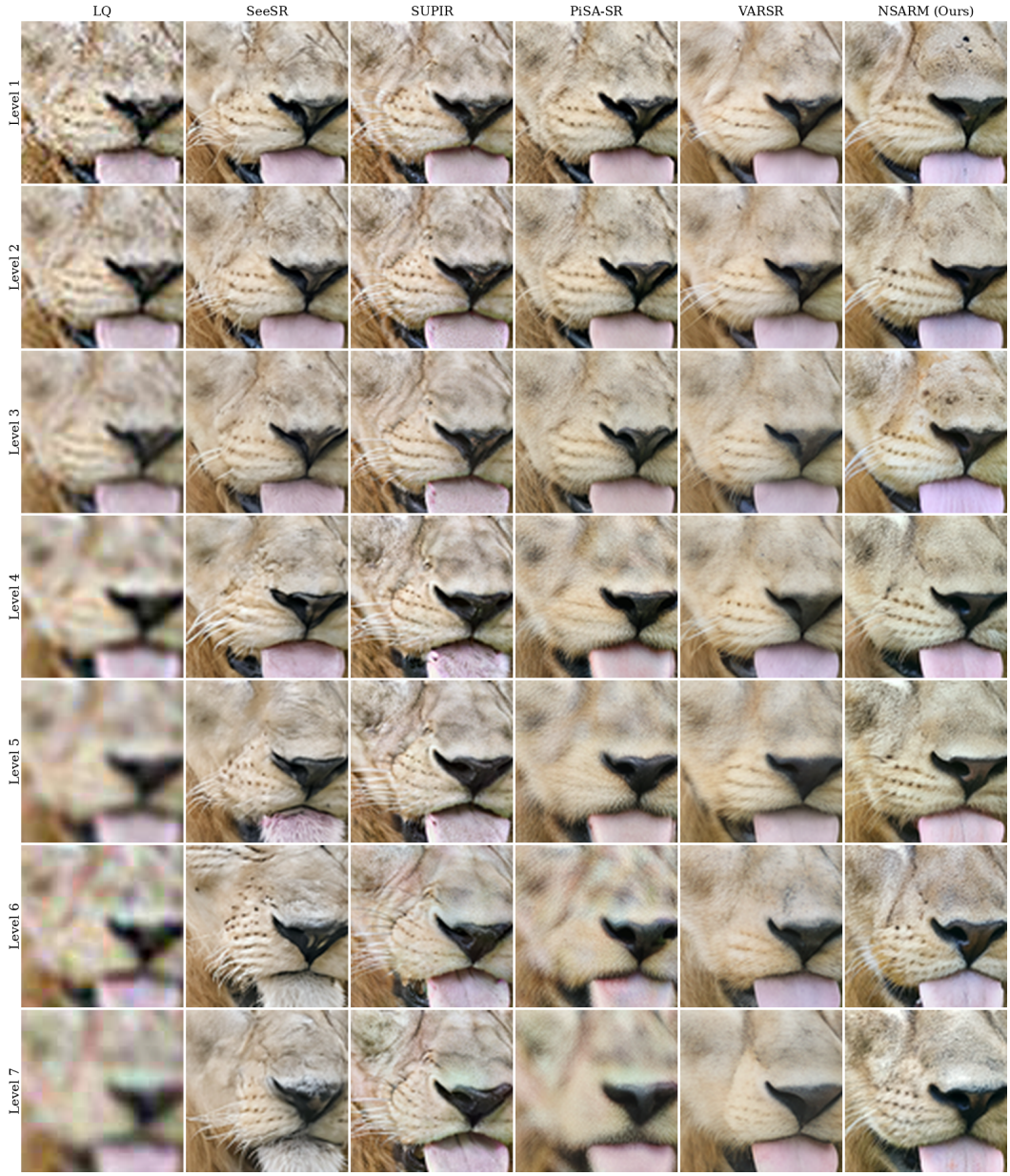



Figure D.8: The visual results of models under different degradation levels. Levels 1–5: within all methods' training degradations; Level 6: Maximum degradation of original RealESRGAN setting; Level 7: setting beyond all methods' training degradations.

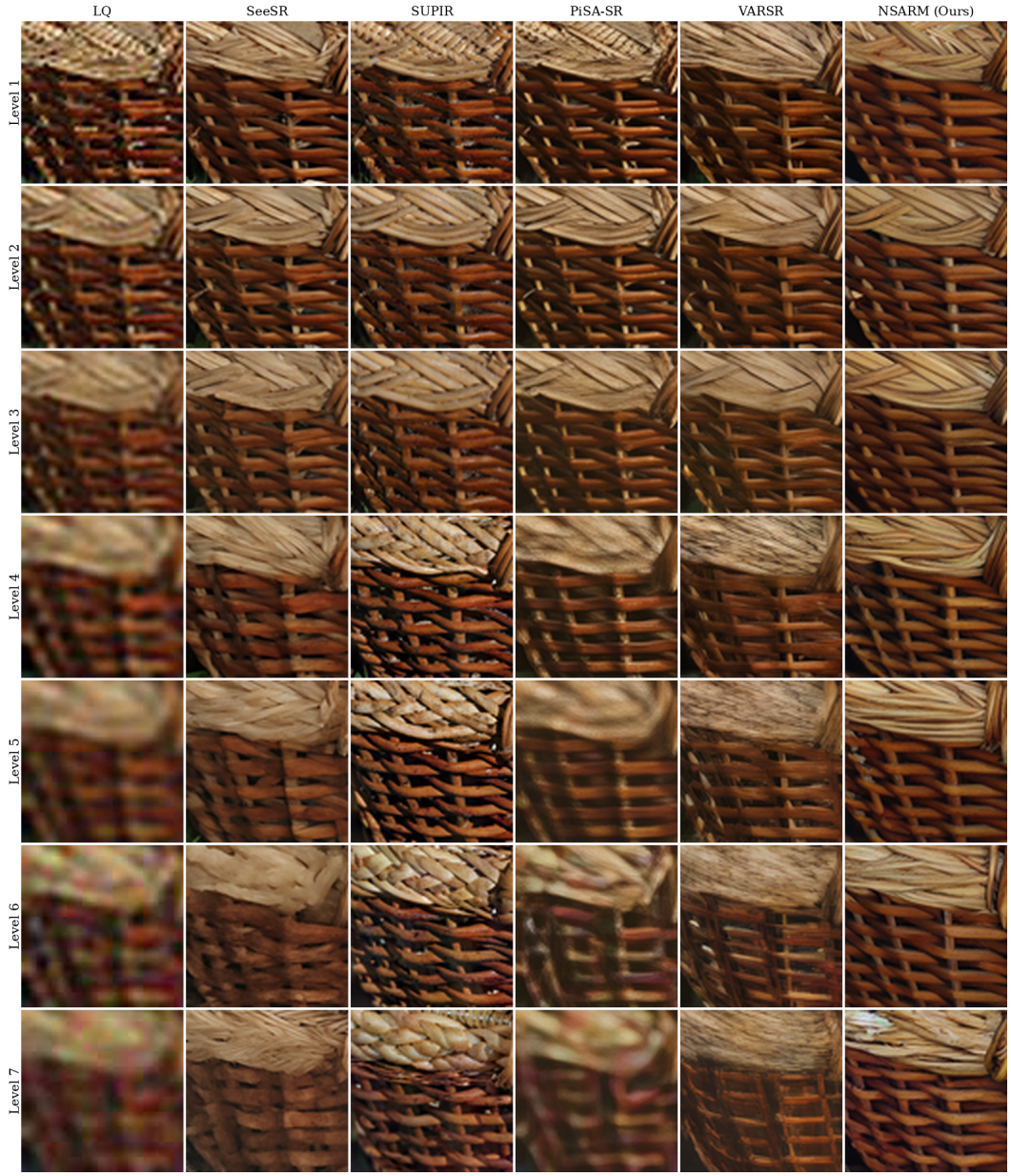



Figure D.9: The visual results of models under different degradation levels. Levels 1–5: within all methods' training degradations; Level 6: Maximum degradation of original RealESRGAN setting; Level 7: setting beyond all methods' training degradations.

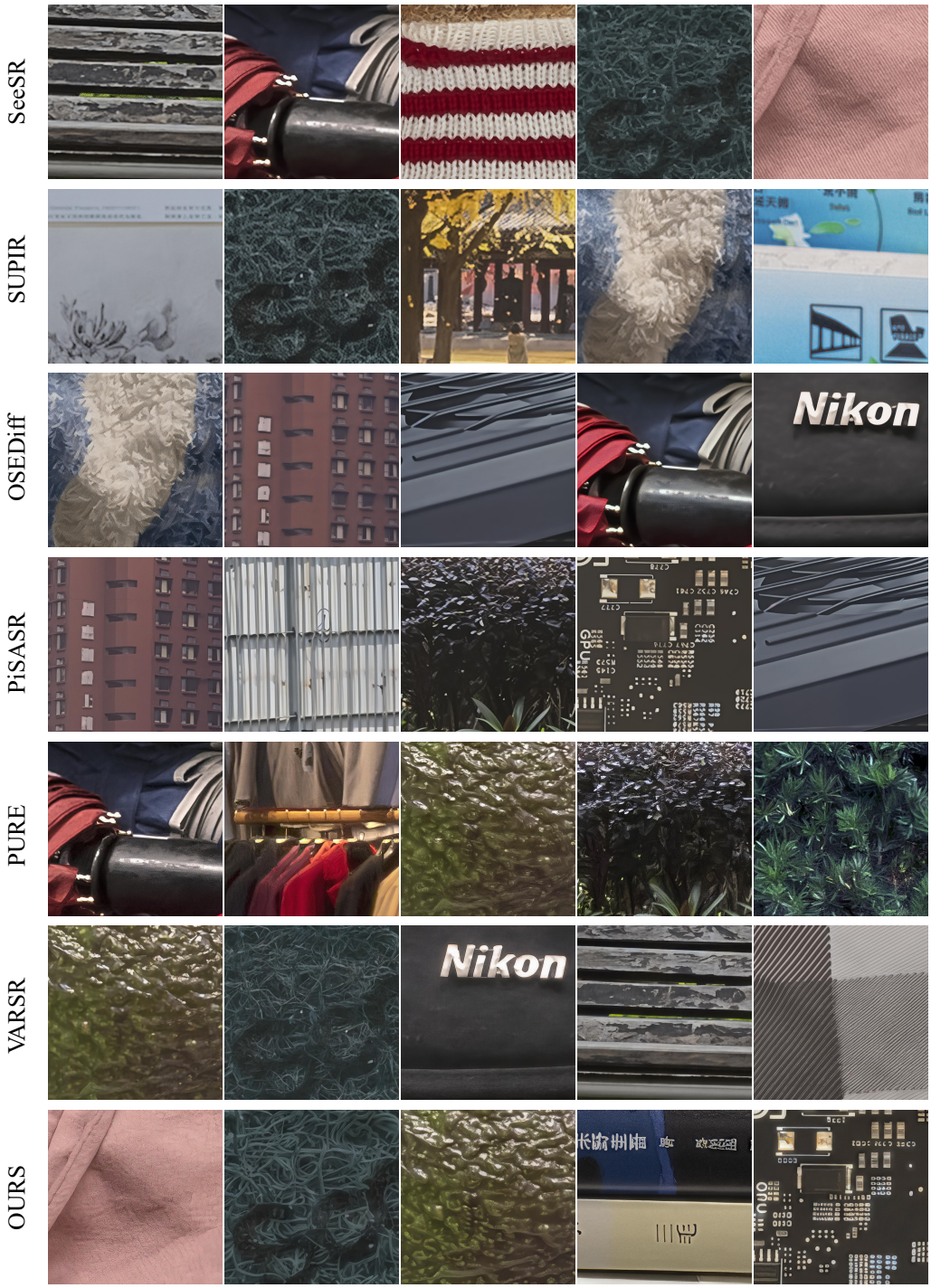



Figure D.10: The visual results of different methods with the worst perceptual metrics on RealSR testset.

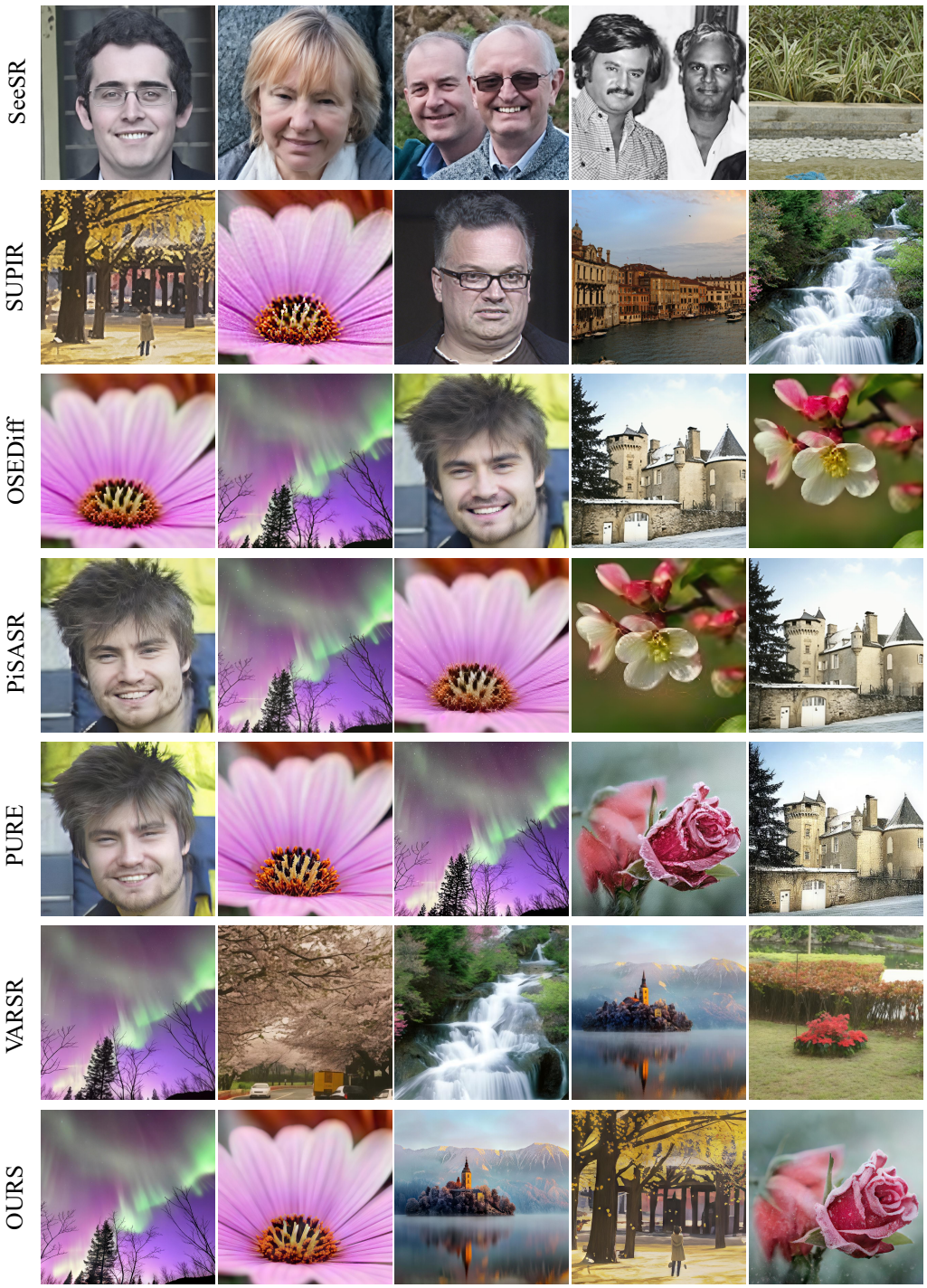



Figure D.11: The visual results of different methods with the worst perceptual metrics on RP60 testset.




RealSR



RP60

Figure D.12: The images that NSARM surpasses the others the most in terms of perception metrics.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are stated in the abstract and introduction, including the proposed NSARM framework, its two-stage training strategy, and its robustness and efficiency advantages, and these claims are supported by the experiments and analyses in the main paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated limitations paragraph in the conclusion, discussing the main limitation that NSARM has a relatively large model size.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: Not applicable. The paper does not present theoretical results requiring formal assumptions and proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the model pipeline, two-stage training strategy, datasets, degradation pipeline, compared methods, evaluation protocol, and key training details needed to understand and reproduce the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper does not provide open access to the training code, model checkpoints, or full reproduction scripts at submission time.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper reports the optimizer, learning rates, batch sizes, iteration numbers, training data source, degradation synthesis, target resolutions, testing datasets, and evaluation metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper does not report confidence intervals, error bars, or repeated-run statistics for the experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper reports the use of 16 NVIDIA A800 GPUs and inference time on an A800 80G GPU, but it does not provide full training-time or total compute estimates for each experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: To the best of our knowledge, this research complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper focuses on algorithmic and empirical contributions and does not include a dedicated discussion of broader societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a high-risk model or scraped dataset as part of the submission.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: We adhere to the open-source agreement, use these models and data, and follow the agreement to potentially re-open source them.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Not applicable. The paper does not release new datasets, code, or models as assets at submission time.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [No]

    Justification: The paper reports the setup and aggregate results, but does not include the full participant instructions, screenshots, or compensation details.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

Justification: The paper includes a small user study involving image-quality preference judgments. This study was minimal-risk and did not collect sensitive personal information, but the manuscript does not report IRB approval or an equivalent review.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Not applicable. LLMs are not an important or non-standard component of the core method in this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.